Aggregating Pairwise Semantic Differences for Few-Shot Claim Veracity Classification

Anonymous ACL submission

Abstract

As part of an automated fact-checking pipeline, the claim veracity classification task consists in determining if a claim is supported by an associated piece of evidence. The complexity of gathering labelled claim-evidence pairs leads to a scarcity of datasets, particularly when dealing with new domains. In this paper, we introduce SEED, a novel vector-based method to few-shot claim veracity classification that aggregates pairwise semantic differences for claim-evidence pairs. We build on the hypothesis that we can find class representative vectors that capture average semantic differences for claim-evidence pairs in a class, which can then be used for classification of new instances. We compare the performance of our method with competitive baselines including fine-tuned BERT/RoBERTa models, as well as the state-of-the-art few-shot veracity classification method that leverages language model perplexity. Experiments conducted on the FEVER and SCIFACT datasets show consistent improvements over competitive baselines in few-shot settings. Our code is available here.[1]

1 Introduction

As a means to mitigate the impact of online misinformation, research in automated fact-checking is attracting increasing attention (Zeng et al., 2021). A typical automated fact-checking pipeline consists of two main components: (1) claim detection, which consists in identifying the set of sentences, out of a long text, deemed capable of being fact-checked (Konstantinovskiy et al., 2020), and (2) claim validation, which aims to do both evidence retrieval and veracity classification for claims (Pradeep et al., 2020). As a key component of the automated fact-checking pipeline, the veracity classification component is generally framed as a task in which a model needs to determine if a claim is supported by a given piece of evidence (Hanselowski et al., 2018; Thorne et al., 2018; Wadden et al., 2020; Lee et al., 2021). It is dominantly tackled as a label prediction task: given a claim c and a piece of evidence e, predict the veracity label for the claim c which can be one of "*Support*", "*Contradict*" or "*Neutral*". For example, the claim "A staging area is only an unused piece of land." is contradicted by the evidence "A staging area (otherwise staging point, staging base or staging post) is a location where organisms, people, vehicles, equipment or material are assembled before use."

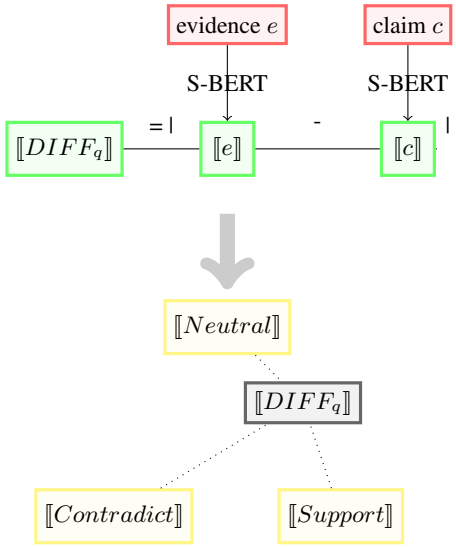

Figure 1: SEED consists of two steps: 1. Captures average semantic differences between claim-evidence pairs for each class, leading to a $[\![DIFF_q]\!]$ representative vector per class. 2. During inference, each input vector $[\![DIFF_q]\!]$ is compared with these representative vectors.

Despite recent advances in the veracity classification task, existing methods predominantly involve training big language models, and/or rely on substantial amounts of labelled data, which can be unrealistic in the case of newly emerging domains such as COVID-19 (Saakyan et al., 2021). To over-

[1] Github repository link is omitted for blind review.

come these dependencies, we set out to propose a novel and effective method to veracity classification with very limited data, e.g. as few as 10 to 20 samples per veracity class. To develop such method, we hypothesise that a method can leverage a small number of training instances, such that the semantic differences within claim-evidence pairs will be similar for each veracity class. Hence, we can calculate a representative vector for each class by averaging semantic differences within claim-evidence pairs of that class. These representative vectors would then enable making predictions on unseen claim-evidence pairs. Figure 1 provides an illustration.

Building on this hypothesis, we propose a novel method, Semantic Embedding Element-wise Difference (SEED), as a method that can leverage a pre-trained language model to build class representative vectors out of claim-evidence semantic differences, which are then used for inference. The method can be flexibly used with any language models, although for experimental purposes here we make use of sentence-BERT (Reimers and Gurevych, 2019). By evaluating on two benchmark datasets –FEVER and SCIFACT–, and comparing both with fine-tuned language models –BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019)– and with the state-of-the-art few-shot claim veracity classification method that leverages perplexity (Lee et al., 2021), we demonstrate the effectiveness of our method. SEED validates the effectiveness of our proposed paradigm to tackle the veracity classification task based on semantic differences, which we consistently demonstrate in three different settings on two datasets.

We make the following contributions:

- We introduce SEED, a novel method that computes semantic differences within claim-evidence pairs for effective and efficient few-shot claim veracity classification.
- By experimenting on two datasets, we demonstrate the effectiveness of SEED to outperform two competitive baselines in the most challenging settings with limited numbers of shots. While the state-of-the-art perplexity-based model is restricted to two-class classification, SEED offers the flexibility to be used in two- or three-class settings.
- We perform a post-hoc analysis of the method, further delving into the results to understand performance variability through standard deviations, as well as to understand method convergence through the evolution of representative vectors.

2 Related Work

The recent increase of interest in automated fact-checking research is evident in survey papers covering different angles: Thorne and Vlachos (2018) focuses on unifying the task formulations and methodologies, Kotonya and Toni (2020b) centers on generating explanations, Nakov et al. (2021) elaborates on assisting human fact checkers, Zeng et al. (2021) overviews the emerging tasks of claim detection and claim validation, and finally Guo et al. (2021) presents a comprehensive and up-to-date survey that highlights research challenges. Publicly available datasets have been gradually improving in terms of scale (Thorne et al., 2018; Sathe et al., 2020; Aly et al., 2021), enriched features (Augenstein et al., 2019; Ostrowski et al., 2020; Kotonya and Toni, 2020a), on-demand domains (Wadden et al., 2020; Diggelmann et al., 2021; Saakyan et al., 2021), and novel perspectives (Chen et al., 2019; Schuster et al., 2021). Recently proposed systems address various challenges, e.g. improving evidence retrieval in a noisy setting (Samarinas et al., 2021), understanding the impact of evidence-aware sentence selection (Bekoulis et al., 2021), developing domain-transferable fact verification (Mithun et al., 2021).

When dealing with veracity classification, most recent systems fine-tune a large pre-trained language model to do three-way label prediction, including VERISCI (Wadden et al., 2020), VERT5ERINI (Pradeep et al., 2020), ParagraphJoint (Li et al., 2021). Despite the evident effectiveness of these methods, fine-tuning models depends on the availability of substantial amounts of labelled data, which are not always accessible, particularly for new domains. They can also be very demanding in terms of computing resources and time. Given these limitations, here we argue for the need of developing more affordable solutions which can in turn achieve competitive performance in few-shot settings and/or with limited computing resources.

Research in few-shot veracity classification is however still in its infancy. To the best of our knowledge, existing work has limited its applicability to binary veracity classification, i.e., keeping the *"Support"* class and merging the *"Contradict"*

and *"Neutral"* classes into a new *"Not_Support"* class. Lee et al. (2021) hypothesised that evidence-conditioned perplexity score from language models would be helpful for assessing claim veracity. They explored using perplexity scores with a threshold th to determine claim veracity into *"Support"* and *"Not_Support"*: if the score is lower than the threshold th, it is classified as *"Not_Support"* and otherwise *"Support"*. This method proved to achieve better performance on few-shot binary classification than fine-tuning a BERT model. In proposing our SEED method, we use this method as the state-of-the-art baseline for few-shot veracity classification in the same two-class settings, while SEED is also applicable to and experimented in three-class settings.

Use of class representative vectors for text classification has also attracted interest in the research community recently. In a similar vein to our proposed approach SEED, prototypical networks (Snell et al., 2017) have proven successful in few-shot classification as a method using representative vectors for each class in classification tasks. Prototypical networks were proposed as a solution to iteratively build class prototype vectors for image classification through parameter updates via stochastic gradient descent, and have recently been used for relation extraction in NLP (Gao et al., 2019; Fu and Grishman, 2021). While building on a similar idea, our SEED method further proposes the use of semantic differences to come up with a meaningful and comparable representation of claim-evidence pairs, a method that has not been studied in the context of claim veracity classification.

3 SEED: Methodology

We hypothesise that we can make use of sentence embeddings (Reimers and Gurevych, 2019) from pre-trained language models such as BERT and RoBERTa to effectively compute pairwise semantic differences between claims and their associated evidences. These differences can then be averaged into a representative vector for each class, which can in turn serve to make predictions on unseen instances during inference.

We formalise this hypothesis through the implementation of SEED as follows. For a given claim-evidence pair made of claim c and evidence e, we first leverage a pre-trained language model to obtain sentence embeddings $[\![c]\!]$ and $[\![e]\!]$. We then capture a representation of their semantic difference by calculating the element-wise difference $|[\![c]\!] - [\![e]\!]|$, following the method proposed by Reimers and Gurevych (2019) as part of the classification objective function. Formally, for a claim-evidence pair x that has $sentence_{x_c}$ and $sentence_{x_e}$, we have:

$$[\![DIFF_x]\!] = |[\![sentence_{x_c}]\!] - [\![sentence_{x_e}]\!]| \quad (1)$$

To address the task of veracity classification that compares a claim with its corresponding evidence, we obtain the mean vector of all $[\![DIFF]\!]$ vectors within a class. We store this mean vector as the representative of the target claim-evidence relation. That is, for each class c that has n training samples available, we obtain its representative relation vector with equation 2.

$$\begin{aligned} &[\![Relation_c]\!] \\ &= \overline{[\![DIFF_c]\!]} \\ &= \frac{1}{n}\sum_{i=1}^{n}([\![DIFF_i]\!]) \\ &= \frac{1}{n}\sum_{i=1}^{n}(|[\![evidence_i]\!] - [\![claim_i]\!]|) \end{aligned} \quad (2)$$

During inference, we first obtain the query $[\![DIFF_q]\!]$ vector for a given unseen claim-evidence pair, then calculate Euclidean distance between the $[\![DIFF_q]\!]$ vector and every computed $[\![Relation_c]\!]$ vector, e.g. $[\![Support]\!]$, $[\![Contradict]\!]$ and $[\![Neutral]\!]$ for three-way veracity classification, and finally inherit the veracity label from the candidate relation vector that has the smallest Euclidean distance value.

4 Experiment Settings

4.1 Datasets

We conduct experiments on the FEVER (Thorne et al., 2018) and SCIFACT (Wadden et al., 2020) datasets (see examples in Table 1). FEVER, a benchmark, large-scale dataset for automated fact-checking, contains claims that are manually modified from Wikipedia sentences and their corresponding Wikipedia evidences. SCIFACT is a smaller dataset that focuses on scientific claims. The claims are annotated by experts and evidences are retrieved from research paper abstracts. For notation consistency, we use *"Support"*, *"Contradict"* and *"Neutral"* as veracity labels for both datasets.[2]

[2]Originally, FEVER uses *"Support"*, *"Refute"* and *"Not Enough Info"* as veracity categories, while SCIFACT uses

FEVER		
Claim	**Evidence**	**Veracity**
"In 2015, among Americans, more than 50% of adults had consumed alcoholic drink at some point."	"For instance, in 2015, among Americans, 89% of adults had consumed alcohol at some point, 70% had drunk it in the last year, and 56% in the last month."	*"Suppport"*
"Dissociative identity disorder is known only in the United States of America."	"DID is diagnosed more frequently in North America than in the rest of the world, and is diagnosed three to nine times more often in females than in males."	*"Contradict"*
"Freckles induce neuromodulation."	"Margarita Sharapova (born 15 April 1962) is a Russian novelist and short story writer whose tales often draw on her former experience as an animal trainer in a circus."	*"Neutral"*
SCIFACT		
Claim	**Evidence**	**Veracity**
"Macropinocytosis contributes to a cell's supply of amino acids via the intracellular uptake of protein."	"Here, we demonstrate that protein macropinocytosis can also serve as an essential amino acid source."	*"Suppport"*
"Gene expression does not vary appreciably across genetically identical cells."	"Genetically identical cells sharing an environment can display markedly different phenotypes."	*"Contradict"*
"Fz/PCP-dependent Pk localizes to the anterior membrane of notochord cells during zebrafish neuralation."	"These results reveal a function for PCP signalling in coupling cell division and morphogenesis at neurulation and indicate a previously unrecognized mechanism that might underlie NTDs."	*"Neutral"*

Table 1: Veracity classification samples from the FEVER and SCIFACT datasets.

4.2 Method implementation

We implement SEED by using sentence-BERT (Reimers and Gurevych, 2019) with huggingface model hub (Wolf et al., 2020). Specifically, we use three variants of BERT (Devlin et al., 2019) as the base model: BERT-base, BERT-large and BERT-nli. The first two are available on huggingface model hub with model id *bert-base-uncased* and *bert-large-uncased*. The last one has been fine-tuned on natural language inference (NLI) tasks and is available on sentence BERT repository with model id *bert-base-nli-mean-tokens*. We include experiments with $SEED_{BERT_{NLI}}$ due to the proximity between the veracity classification and natural language inference tasks. We use $SEED_{BERT_B}$, $SEED_{BERT_L}$ and $SEED_{BERT_{NLI}}$ to denote them hereafter.

4.3 Baselines

We compare our method with two baseline methods: perplexity-based (PB) method and fine-tuning (FT) method.

Perplexity-Based Method (PB) The perplexity-based method (Lee et al., 2021) is the current SOTA method for few-shot veracity classification. It uses conditional perplexity scores generated by pre-trained language models to find a threshold that enables binary predictions. If the perplexity score of a given claim-evidence pair is higher than the threshold, it is assigned the "Support" label; otherwise, the "Not_Support" label. We conduct experiments with BERT-base and BERT-large for direct comparison with other methods. We denote them as PB_{BERT_B} and PB_{BERT_L} hereafter.

Fine-Tuning Method (FT) We also conduct experiments with widely-used model fine-tuning methods. Specifically, we fine-tune vanilla BERT-base, BERT-large, RoBERTa-base and RoBERTa-large models from huggingface model hub (Wolf et al., 2020). The associated model ids are *bert-base-uncased*, *bert-large-uncased*, *roberta-base* and *roberta-large* respectively. Following Lee et al. (2021), we use $5e^{-6}$ for FT_{BERT_B} and $FT_{RoBERTa_B}$ as learning rate and $2e^{-5}$ for FT_{BERT_L} and $FT_{RoBERTa_L}$. All models share the same batch size of 32 and are trained for 10 epochs. We denote them as FT_{BERT_B}, FT_{BERT_L}, $FT_{RoBERTa_B}$ and $FT_{RoBERTa_L}$ hereafter.

4.4 Experimental Design

Experiments are conducted in three different configurations: binary FEVER veracity classification, three-way FEVER veracity classification and three-

"Supports", *"Refutes"* and *"No_Info"*.

way SCIFACT veracity classification. The first configuration is designed to enable direct comparison with the SOTA method (i.e. PB), as it is only designed for doing binary classification.

We conduct N-shot experiments (i.e. those with n training samples per class) with the following choices of n: 2, 4, 6, 8, 10, 20, 30, 40, 50, 100. Note that one may argue that 50-shot and 100-shot are not necessarily few-shot, however we chose to include them to further visualise the trend of methods up to 100 shots. The number of shots n refers to the number of instances, per class, e.g. 2-shot experiments would include 6 instances in total when experimenting with 3 classes. To control for fluctuations in performance scores owing to the randomness of selecting n shots, for each n-shot experiment we use 10 different random seeds ranging from 123 to 132, and we report the mean results. Likewise, due to the variability in performance of the FT method given its non-deterministic nature, we do 5 runs for each setting and report the mean results.

5 Results

We report overall accuracy performance of each task formulation here.

5.1 FEVER Binary Classification

Experiment Setup For binary classification, we use the FEVER data provided by the original authors of the PB method (Lee et al., 2021) for fair comparison. The data contains 3333 *"Support"* instances and 3333 *"Not_Support"* instances.[3] For n-shot setting, we sample n shots –i.e. n instances per class– as the train set, and use the rest –i.e. $3333 - n$ instances per class– as the test set. We present experiments with all three methods (SEED, PB, FT).

Results As shown in Figure 2, SEED achieves the overall best performance in few-shot settings. When given fewer than 10 shots, the accuracy of the FT method remains low at around 50%, which is close to a random guess for a balanced, binary classification task. Meanwhile, PB_{BERT_B}, PB_{BERT_L}, $SEED_{BERT_B}$ and $SEED_{BERT_L}$ achieve similar results at around 57%. In 10-shot, 20-shot and 30-shot settings, SEED outperforms PB method, which in turn outperforms the FT method. In 40-shot and 50-shot setting, FT_{BERT_L} surpasses PB, although FT_{BERT_B}, $FT_{RoBERTa_B}$ and $FT_{RoBERTa_L}$ perform remarkably lower. In the 100-shot setting, FT_{BERT_L} manages to outperform $SEED_{BERT_B}$ and $SEED_{BERT_L}$ and achieves similar performance as $SEED_{BERT_{NLI}}$. FT_{BERT_B}, $FT_{RoBERTa_B}$ and $FT_{RoBERTa_L}$ in the 100-shot setting failed to outperform SEED, despite that $FT_{RoBERTa_L}$ successfully outperformed PB. Overall, SEED with vanilla pre-trained language models outperforms both baselines in 10-shot to 50-shot settings. In addition, SEED with BERT-nli always achieves the best performance up to 100 shots.

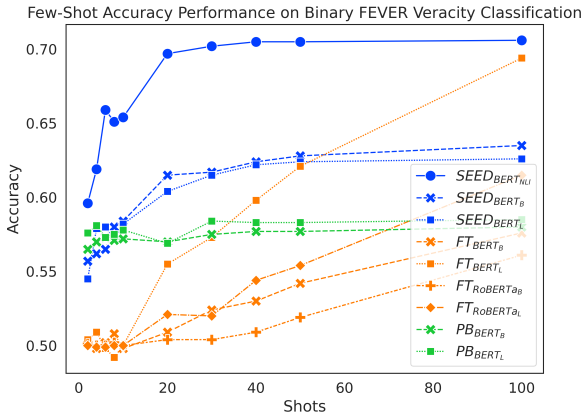

Figure 2: Comparison of few-shot accuracy performance on the binary FEVER dataset.

Interestingly, the increase of shots has very different effects on each method. SEED experiences significant accuracy improvement as shots increase when given fewer than 20 shots; the performance boost then slows down drastically afterwards. Starting with reasonably high accuracy, PB achieves a mild performance improvement when given more training samples. When given fewer than 10 shots, the FT method doesn't experience reliable performance increase over training data increase; it only starts to experience linear performance boost after 10-shots.

5.2 FEVER Three-Way Classification

Experiment Setup We use 3333 randomly sampled instances for each class out of *"Support"*, *"Contradict"* and *"Neutral"* from the original FEVER test set as the total dataset for our experiment. For n-shot setting, we sample n shots, i.e. n instances per class, as the train set, and use the rest, i.e. $3333 - n$ instances per class, as the test set. In these experiments we compare SEED and FT,

[3]The *"Not_Support"* is obtained by sampling and merging original instances from both *"Contradict"* and *"Neutral"*

excluding PB as it cannot be applied to three-class experiments.

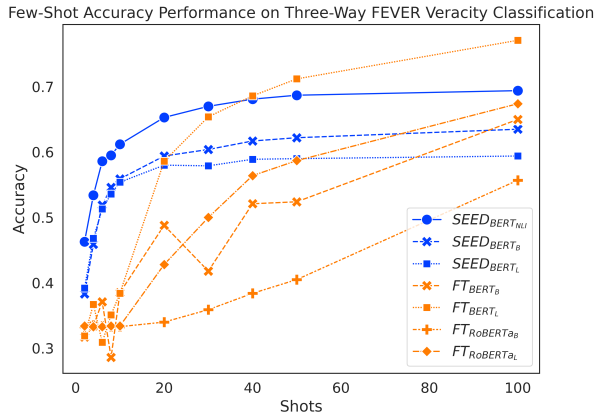

Figure 3: Comparison of few-shot accuracy performance on the FEVER dataset.

Results Figure 3 shows a general trend to increase performance as the amount of training data increases for both methods. When given 10 or fewer shots, SEED shows significant performance advantages. When given between 2 and 10 shots, performance of fine-tuned models fluctuates around 33%, which equals to a random guess. Meanwhile, SEED achieves significant accuracy improvement from less than 40% to around 55% with vanilla pre-trained language models. In this scenario, the performance gap between the two methods that use the same model base ranges from 6% to 26%. With 20 shots, SEED with vanilla pre-trained language models significantly outperform FT_{BERT_B}, $FT_{RoBERTa_B}$ and $FT_{RoBERTa_L}$, although FT_{BERT_L} managed to achieve similar results. With 30 shots, SEED with vanilla pre-trained language models reach its performance peak at around 60% and $SEED_{BERT_{NLI}}$ peaks around 68%. Given 30 or more shots, SEED slowly gets surpassed by the FT method. Specifically, FT_{BERT_L} surpasses SEED with vanilla pre-trained language models using 30 shots, while $FT_{RoBERTa_L}$ and FT_{BERT_B} only achieve a similar effect with 100 shots. However, $FT_{RoBERTa_B}$ never outperforms SEED within 100 shots. In addition, $SEED_{BERT_{NLI}}$ has enormous performance advantages when given fewer than 10 shots, despite being outperformed by FT_{BERT_L} at 40 shots. Overall, SEED experiences a performance boost with very few shots, whereas the FT method is more demanding, whose performance starts to increase only after 10 shots.

Interestingly, $SEED_{BERT_B}$ outperforms $SEED_{BERT_L}$ starting from 6 shots. This performance difference within SEED further results in another interesting observation: $SEED_{BERT_B}$ achieves better overall accuracy than FT_{BERT_L} at 10 shots.

5.3 SCIFACT Three-Way Classification

Experiment Setup The SCIFACT dataset is much smaller than the FEVER dataset, originally with only 809 claims for training and 300 claims for development (the test set being withheld for a shared task is not yet available at the time of writing). For each n-shot setting, we randomly sample n instances for each class out of *"Support"*, *"Contradict"* and *"Neutral"*, which are used as the train set. Given the imbalanced nature of the development set (i.e. 138, 114 and 71 pairs for each class), we randomly sample 70 instances for each class in the development set and use them for evaluation. We again compare SEED and FT in these experiments.

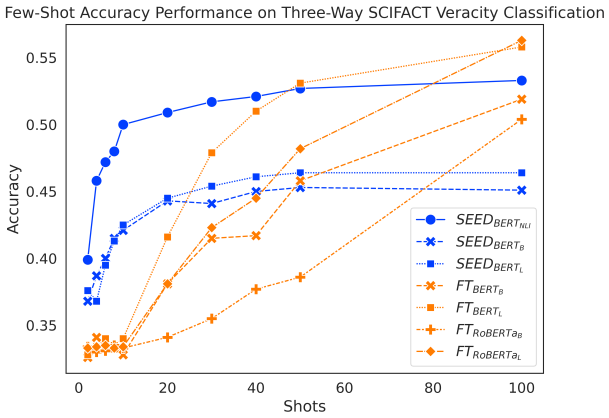

Figure 4: Comparison of few-shot accuracy performance on the SCIFACT dataset.

Results Figure 4 shows again an expected increase in performance for both methods as they use more training data. Despite taking a bit longer to pick up, SEED still starts its performance boost early on. Increasing from 2 to 10 shots, SEED gains a substantial increase in performance. In addition, the FT method performs similarly to a random guess at around 33% accuracy when given 10 or fewer shots. When given 20 shots, FT still falls behind SEED, which differs from the trend seen with the FEVER three-way veracity classi-

fication. $SEED_{BERT_B}$ and $SEED_{BERT_L}$ peak at around 45%, while $SEED_{BERT_{NLI}}$ peaks at around 50% with only 20 shots. At 30-shots and 40-shots, SEED still shows competitive performance, where FT_{BERT_L} outperforms two of the SEED variants, but still falls behind $SEED_{BERT_{NLI}}$. $FT_{RoBERTa_L}$ outperforms SEED with vanilla BERT models at 50-shots and FT_{BERT_B} and $FT_{RoBERTa_B}$ achieves that at 100-shots.

The accuracy scores on the SCIFACT dataset are noticeably lower than on the FEVER dataset. The FT method is again more demanding on the number of shots and experiences a noticeable delay to overtake SEED, more so on SCIFACT than on FEVER. This highlights the challenging nature of the SCIFACT dataset, where SEED still remains the best in few-shot settings.

6 Post-hoc Analysis

6.1 Impact of shot sampling on performance

Random selection of n shots for few-shot experiments can lead to a large variance in the results, which we mitigate by presenting averaged results for 10 samplings. To further investigate the variability of the three methods under study, we look into the standard deviations.

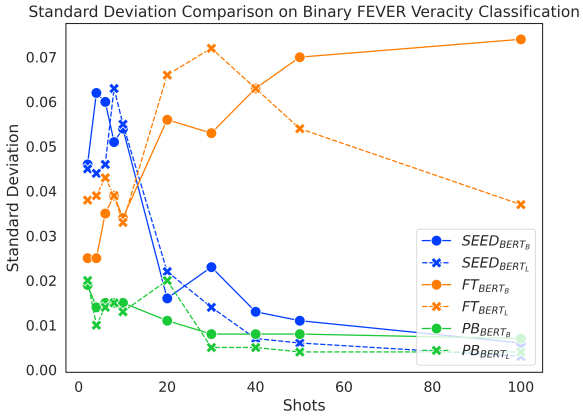

Figure 5: Standard deviation comparison on binary FEVER veracity classification.

Figure 5 presents the standard deviation distribution on Binary FEVER Veracity Classification, which is largely representative of the standard deviations of the models across the different settings (for detailed standard deviation values across settings please refer to Appendix C.). We only analyse configurations that utilise BERT-base and BERT-large for direction comparison across methods. Overall, PB always has the lowest standard deviation, which demonstrates its low performance variability across random sampling seeds. When given 10 or fewer shots, the standard deviation of SEED is comparatively higher than that of FT. It implies that the SEED method experiences larger performance fluctuations when given very few shots. Despite its robustness to random sampling when given 10 or fewer shots, FT's accuracy performance remains significantly lower than other methods and close to random guess, as shown in Figure 2. Furthermore, when given more than 10 shots, the standard deviations of FT surpass SEED with large margin. The FT method loses its advantages in robustness and becomes more vulnerable to random sampling than the SEED method.

In short, PB is the most robust method to sample variations, despite underperforming SEED on average; SEED is still generally more robust than the FT method, except for cases with fewer shots where FT underperforms.

6.2 Why does SEED plateau?

As presented in §5, the performance improvement of SEED becomes marginal when given more than 40 shots. Given that SEED learns mean representative vectors based on training instances for each class, the method likely reaches a stable average vector after seeing a number of shots. To investigate the converging process of representative vectors, we measure the variation caused in the mean vectors by each additional shot added. Specifically, for values of n ranging from 2 to 200, we calculate the Euclidean distance between n-shot relation vectors and (n-1)-shot representative vectors, which measures the extent to which representative vectors were altered since the addition of the last shot. Figure 6 depicts the converging process with FEVER three-way veracity classification. Across three different model bases, the amount of variation drops consistently for larger numbers of n shots, with a more prominent drop for n={2-21} and a more modest drop subsequently. From a positive angle, this indicates the ability of SEED to converge quickly, which validates the use of semantic differences for verification. From a negative angle, it also means that the method stops learning as much for larger numbers of shots as it becomes stable.

The curves of BERT-base and BERT-large largely overlap each other, while the curve of BERT-nli does not conjoin until convergence. It

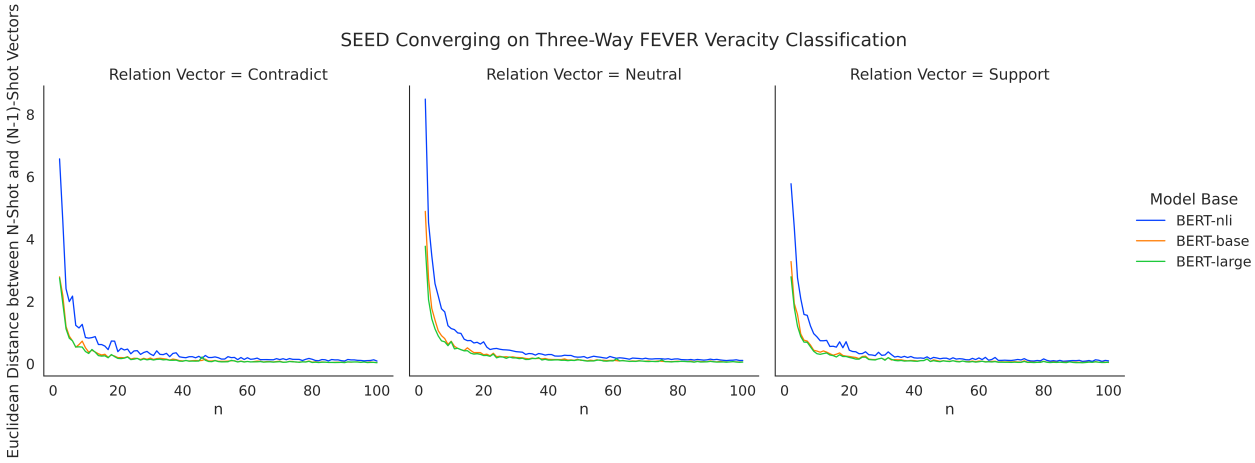

Figure 6: SEED converging on three-way FEVER veracity classification with increasing number of n shots.

corresponds well with the overall performance advantages of utilising BERT-nli as presented in §5. It implies that using language models fine-tuned on relevant tasks allow larger impact to be made with initial few shots. Future work may deepen the explorations in this direction. For example, using a model fine-tuned on FEVER veracity classification to address SCIFACT veracity classification.

7 Discussion

With experiments on two- and three-class settings on two datasets, FEVER and SCIFACT, SEED shows state-of-the-art performance in few-shot settings. With only 10 shots, SEED with vanilla BERT models achieves approximately 58% accuracy on binary veracity classification, 8% above FT and 1% above PB. Furthermore, SEED achieves around 56% accuracy on three-way FEVER, while FT models underperform with a 38% accuracy, an absolute performance gap of 18%. Despite the difficulty of performing veracity classification on scientific texts in the SCIFACT dataset, SEED still achieves accuracy above 42%, which is 9% higher than FT. When utilising BERT-nli, SEED consistently achieves improvements with 10 shots only: 15% higher than FT and 8% higher than PB on FEVER binary veracity classification; 23% higher than FT on FEVER three-way veracity classification and 17% higher than FT on SCIFACT three-way veracity classification. Further, Appendix A presents detailed classwise F1 performance, which shows that improved performance is also consistent across classes.

In comparison with PB, SEED has better learning capacities, higher few-shot performance, and most importantly, it is more flexible for doing multi-way veracity classification, enabling in this case both two-class and three-class experiments. With respect to FT, SEED is better suited and faster to deploy in few-shot settings. It is more effective regarding few-shot data usage, generally more robust to random sampling, and it has lower demand on data quantity and computing resources.

While SEED demonstrates the ability to learn representative vectors that lead to effective veracity classification with limited labelled data and computational resources, its performance plateaus with large numbers of shots. SEED has proven effective for few-shot claim veracity classification experiments. Its extension to adapt to scenarios with more shots remains an open problem that is beyond the scope of this work.

8 Conclusions

We have presented an efficient and effective SEED method which achieves significant improvements over the baseline systems in few-shot veracity classification. By comparing it with a perplexity-based few-shot claim veracity classification method as well as a range of fine-tuned language models, SEED achieves state-of-the-art performance in the task on two datasets and three different settings. Given its low demand on labelled data and computational resources, SEED can be easily extended, for example, to new domains with limited labelled examples.

Acknowledgements

Omitted for blind review.

References

Rami Aly, Zhijiang Guo, Michael Schlichtkrull, James Thorne, Andreas Vlachos, Christos Christodoulopoulos, Oana Cocarascu, and Arpit Mittal. 2021. FEVEROUS: Fact Extraction and VERification Over Unstructured and Structured information. *arXiv:2106.05707 [cs]*. ArXiv: 2106.05707.

Isabelle Augenstein, Christina Lioma, Dongsheng Wang, Lucas Chaves Lima, Casper Hansen, Christian Hansen, and Jakob Grue Simonsen. 2019. MultiFC: A Real-World Multi-Domain Dataset for Evidence-Based Fact Checking of Claims. *arXiv:1909.03242 [cs, stat]*. ArXiv: 1909.03242.

Giannis Bekoulis, Christina Papagiannopoulou, and Nikos Deligiannis. 2021. Understanding the Impact of Evidence-Aware Sentence Selection for Fact Checking. In *Proceedings of the Fourth Workshop on NLP for Internet Freedom: Censorship, Disinformation, and Propaganda*, pages 23–28, Online. Association for Computational Linguistics.

Sihao Chen, Daniel Khashabi, Wenpeng Yin, Chris Callison-Burch, and Dan Roth. 2019. Seeing Things from a Different Angle:Discovering Diverse Perspectives about Claims. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 542–557, Minneapolis, Minnesota. Association for Computational Linguistics.

J. Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL-HLT*.

Thomas Diggelmann, Jordan Boyd-Graber, Jannis Bulian, Massimiliano Ciaramita, and Markus Leippold. 2021. CLIMATE-FEVER: A Dataset for Verification of Real-World Climate Claims. *arXiv:2012.00614 [cs]*. ArXiv: 2012.00614.

Lisheng Fu and Ralph Grishman. 2021. Learning Relatedness between Types with Prototypes for Relation Extraction. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2011–2016, Online. Association for Computational Linguistics.

Tianyu Gao, Xu Han, Zhiyuan Liu, and Maosong Sun. 2019. Hybrid Attention-Based Prototypical Networks for Noisy Few-Shot Relation Classification. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):6407–6414. Number: 01.

Zhijiang Guo, Michael Schlichtkrull, and Andreas Vlachos. 2021. A Survey on Automated Fact-Checking. *arXiv:2108.11896 [cs]*. ArXiv: 2108.11896.

Andreas Hanselowski, Hao Zhang, Zile Li, Daniil Sorokin, Benjamin Schiller, Claudia Schulz, and Iryna Gurevych. 2018. UKP-Athene: Multi-Sentence Textual Entailment for Claim Verification. In *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*, pages 103–108, Brussels, Belgium. Association for Computational Linguistics.

Lev Konstantinovskiy, Oliver Price, Mevan Babakar, and Arkaitz Zubiaga. 2020. Towards Automated Factchecking: Developing an Annotation Schema and Benchmark for Consistent Automated Claim Detection. *arXiv:1809.08193 [cs]*. ArXiv: 1809.08193.

Neema Kotonya and Francesca Toni. 2020a. Explainable Automated Fact-Checking: A Survey. *arXiv:2011.03870 [cs]*. ArXiv: 2011.03870.

Neema Kotonya and Francesca Toni. 2020b. Explainable Automated Fact-Checking for Public Health Claims. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7740–7754, Online. Association for Computational Linguistics.

Nayeon Lee, Yejin Bang, Andrea Madotto, and Pascale Fung. 2021. Towards Few-shot Fact-Checking via Perplexity. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1971–1981, Online. Association for Computational Linguistics.

Xiangci Li, Gully Burns, and Nanyun Peng. 2021. A Paragraph-level Multi-task Learning Model for Scientific Fact-Verification. *arXiv:2012.14500 [cs]*. ArXiv: 2012.14500.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv:1907.11692 [cs]*. ArXiv: 1907.11692.

Mitch Paul Mithun, Sandeep Suntwal, and Mihai Surdeanu. 2021. Data and Model Distillation as a Solution for Domain-transferable Fact Verification. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4546–4552, Online. Association for Computational Linguistics.

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:2103.07769 [cs]*. ArXiv: 2103.07769.

Wojciech Ostrowski, Arnav Arora, Pepa Atanasova, and Isabelle Augenstein. 2020. Multi-Hop Fact Checking of Political Claims. *arXiv:2009.06401 [cs]*. ArXiv: 2009.06401.

Ronak Pradeep, Xueguang Ma, Rodrigo Nogueira, and Jimmy Lin. 2020. Scientific Claim Verification with VERT5ERINI. *arXiv:2010.11930 [cs]*. ArXiv: 2010.11930.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. *arXiv:1908.10084 [cs]*. ArXiv: 1908.10084.

Arkadiy Saakyan, Tuhin Chakrabarty, and Smaranda Muresan. 2021. COVID-Fact: Fact Extraction and Verification of Real-World Claims on COVID-19 Pandemic. *arXiv:2106.03794 [cs]*. ArXiv: 2106.03794.

Chris Samarinas, Wynne Hsu, and Mong Li Lee. 2021. Improving Evidence Retrieval for Automated Explainable Fact-Checking. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Demonstrations*, pages 84–91, Online. Association for Computational Linguistics.

Aalok Sathe, Salar Ather, Tuan Manh Le, Nathan Perry, and Joonsuk Park. 2020. Automated Fact-Checking of Claims from Wikipedia. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 6874–6882, Marseille, France. European Language Resources Association.

Tal Schuster, Adam Fisch, and Regina Barzilay. 2021. Get Your Vitamin C! Robust Fact Verification with Contrastive Evidence. *arXiv:2103.08541 [cs]*. ArXiv: 2103.08541.

Jake Snell, Kevin Swersky, and Richard S. Zemel. 2017. Prototypical Networks for Few-shot Learning. *arXiv:1703.05175 [cs, stat]*. ArXiv: 1703.05175.

James Thorne and Andreas Vlachos. 2018. Automated Fact Checking: Task formulations, methods and future directions. *arXiv:1806.07687 [cs]*. ArXiv: 1806.07687.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. FEVER: a large-scale dataset for Fact Extraction and VERification. *arXiv:1803.05355 [cs]*. ArXiv: 1803.05355.

David Wadden, Shanchuan Lin, Kyle Lo, Lucy Lu Wang, Madeleine van Zuylen, Arman Cohan, and Hannaneh Hajishirzi. 2020. Fact or Fiction: Verifying Scientific Claims. *arXiv:2004.14974 [cs]*. ArXiv: 2004.14974.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. HuggingFace's Transformers: State-of-the-art Natural Language Processing. *arXiv:1910.03771 [cs]*. ArXiv: 1910.03771.

Xia Zeng, Amani S. Abumansour, and Arkaitz Zubiaga. 2021. Automated fact-checking: A survey. *Language and Linguistics Compass*, 15(10):e12438. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/lnc3.12438.

A Classwise F1 Performances

We present classwise F1 performance here for further understanding of the results. Figure 7 sheds light on addressing the task of FEVER binary veracity classification. Both SEED and FT method gain improved performance on both classes with more data. The SEED method and PB method have significant performance advantages on the "Support" class, when given 10 or fewer shots. Despite that the PB method initially achieves very high performance on the "Support" class at around 60%, it then experiences a performance drop and ends at around 55% for BERT-base and 58% for BERT-large.

Figures 8 and 9 show consistent classwise performance patterns in tackling three-way veracity classification on both FEVER and SCIFACT. Both figures indicate that SEED has better overall performance in all three classes when given fewer than 20 shots, where performance on the "Support" class always has absolute advantages over the FT method and performance on the "Neutral" class experiences the biggest boost. At around 20-shot,s the FT method starts to overtake largely due to improved performance on the "Neutral" class. Interestingly, within SEED, $SEED_{BERT_B}$ outperforms $SEED_{BERT_L}$, which in turn outperforms $SEED_{BERT_{NLI}}$.

In general, classwise F1 performance shows consistent performance patterns with overall accuracy performance. The SEED method has significant performance advantages when given 10 or fewer shots in all classes. The PB method has very good performance on predicting the "Support" class initially but struggles to improve with more data. The FT method has underwhelming performance on all classes when given very few shots and gain big improvements over training data increase, especially on the "Neutral" class.

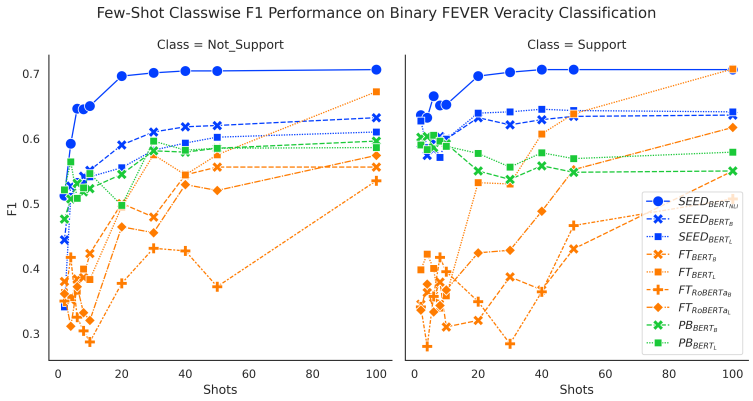

Figure 7: Comparison of few-shot classwise F1 performance on the binary FEVER dataset.

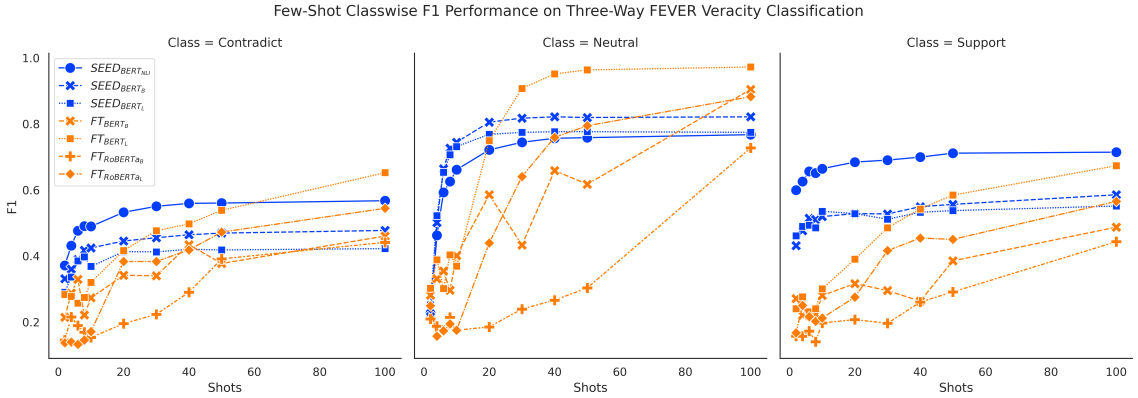

Figure 8: Comparison of few-shot classwise F1 performance on the FEVER dataset.

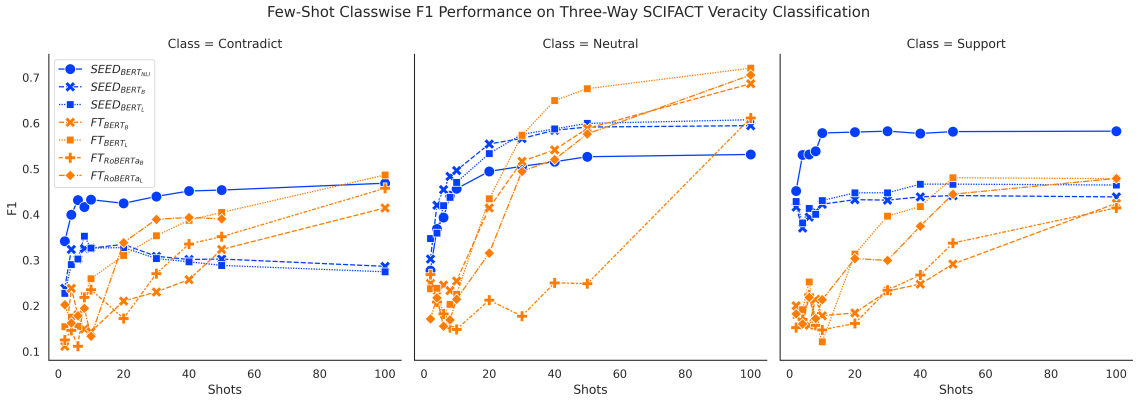

Figure 9: Comparison of few-shot classwise F1 performance on the SCIFACT dataset.

B Detailed Accuracy and Classwise F1 Scores

We report detailed performance scores of the three conducted experiments here, namely FEVER binary veracity classification, FEVER three-way veracity classification and SCIFACT three-way veracity classification. All of the reported scores are mean scores of multiple runs.

B.1 FEVER Binary Veracity Classification

Table 2 reports detailed few-shot performance for PB_{BERT_B} and PB_{BERT_L}. Table 3 reports detailed few-shot performance for FT_{BERT_B}, FT_{BERT_L}, $FT_{RoBERTa_B}$ and $FT_{RoBERTa_L}$. Table 4 reports detailed few-shot performance for $SEED_{BERT_B}$, $SEED_{BERT_L}$ and $SEED_{BERT_{NLI}}$.

	PB_{BERT_B}			PB_{BERT_L}		
Shots	Acc	$F1_S$	$F1_{Not}$	Acc	$F1_S$	$F1_{Not}$
2	0.565	0.602	0.476	0.576	0.590	0.521
4	0.570	0.603	0.507	0.581	0.583	0.564
6	0.573	0.586	0.531	0.573	0.605	0.508
8	0.571	0.594	0.518	0.575	0.596	0.524
10	0.572	0.592	0.523	0.578	0.588	0.546
20	0.570	0.550	0.545	0.569	0.577	0.497
30	0.575	0.537	0.581	0.584	0.556	0.596
40	0.577	0.558	0.579	0.583	0.578	0.582
50	0.577	0.548	0.585	0.583	0.569	0.585
100	0.580	0.550	0.596	0.585	0.579	0.586

Table 2: Few-Shot PB Performance on FEVER Binary Veracity Classification. Acc stands for accuracy; $F1_S$, $F1_{Not}$ stands for F1 score for "Support" and "Not_Support" respectively.

	FT_{BERT_B}			FT_{BERT_L}		
Shots	Acc	$F1_S$	$F1_{Not}$	Acc	$F1_S$	$F1_{Not}$
2	0.501	0.345	0.380	0.504	0.398	0.363
4	0.498	0.363	0.352	0.509	0.422	0.355
6	0.502	0.355	0.384	0.498	0.400	0.365
8	0.508	0.379	0.386	0.492	0.339	0.399
10	0.498	0.310	0.423	0.500	0.358	0.383
20	0.509	0.320	0.500	0.555	0.532	0.495
30	0.524	0.387	0.479	0.573	0.530	0.575
40	0.530	0.367	0.544	0.598	0.607	0.544
50	0.542	0.430	0.556	0.621	0.638	0.575
100	0.576	0.550	0.556	0.694	0.707	0.672
	$FT_{RoBERTa_B}$			$FT_{RoBERTa_L}$		
Shots	Acc	$F1_S$	$F1_{Not}$	Acc	$F1_S$	$F1_{Not}$
2	0.501	0.341	0.350	0.500	0.336	0.361
4	0.500	0.280	0.417	0.499	0.376	0.311
6	0.500	0.357	0.325	0.499	0.333	0.372
8	0.502	0.417	0.304	0.500	0.343	0.332
10	0.500	0.395	0.287	0.500	0.367	0.320
20	0.504	0.349	0.377	0.521	0.424	0.464
30	0.504	0.284	0.431	0.520	0.428	0.455
40	0.509	0.364	0.427	0.544	0.488	0.529
50	0.519	0.466	0.372	0.554	0.552	0.520
100	0.561	0.507	0.535	0.615	0.617	0.574

Table 3: Few-Shot FT Performance on FEVER Binary Veracity Classification. Acc stands for accuracy; $F1_S$, $F1_{Not}$ stands for F1 score for "Support" and "Not_Support" respectively.

B.2 FEVER Three-way Veracity Classification

Table 5 reports detailed few-shot performance for FT_{BERT_B}, FT_{BERT_L}, $FT_{RoBERTa_B}$ and $FT_{RoBERTa_L}$. Table 6 reports detailed few-shot performance for $SEED_{BERT_B}$, $SEED_{BERT_L}$ and $SEED_{BERT_{NLI}}$.

B.3 SCIFACT Three-way Veracity Classification

Table 7 reports detailed few-shot performance for FT_{BERT_B}, FT_{BERT_L}, $FT_{RoBERTa_B}$ and $FT_{RoBERTa_L}$.

Table 8 reports detailed few-shot performance for $SEED_{BERT_B}$, $SEED_{BERT_L}$ and $SEED_{BERT_{NLI}}$.

	$SEED_{BERT_B}$			$SEED_{BERT_L}$		
Shots	Acc	$F1_S$	$F1_{Not}$	Acc	$F1_S$	$F1_{Not}$
2	0.557	0.592	0.444	0.545	0.627	0.341
4	0.562	0.574	0.527	0.579	0.586	0.511
6	0.565	0.583	0.530	0.580	0.593	0.534
8	0.580	0.603	0.542	0.572	0.571	0.531
10	0.584	0.599	0.551	0.582	0.599	0.541
20	0.615	0.632	0.590	0.604	0.639	0.555
30	0.617	0.621	0.610	0.615	0.641	0.582
40	0.624	0.629	0.618	0.622	0.645	0.593
50	0.628	0.634	0.620	0.624	0.643	0.602
100	0.635	0.636	0.632	0.626	0.641	0.610
	$SEED_{BERT_{NLI}}$					
Shots	Acc		$F1_S$		$F1_{Not}$	
2	0.596		0.636		0.512	
4	0.619		0.632		0.592	
6	0.659		0.665		0.646	
8	0.651		0.651		0.645	
10	0.654		0.652		0.650	
20	0.697		0.696		0.696	
30	0.702		0.702		0.701	
40	0.705		0.706		0.704	
50	0.705		0.706		0.704	
100	0.706		0.706		0.706	

Table 4: Few-Shot SEED Performance on FEVER Binary Veracity Classification. Acc stands for accuracy; $F1_S$, $F1_{Not}$ stands for F1 score for "Support" and "Not_Support" respectively.

	FT_{BERT_B}				FT_{BERT_L}			
Shots	Acc	$F1_C$	$F1_N$	$F1_S$	Acc	$F1_C$	$F1_N$	$F1_S$
2	0.317	0.214	0.281	0.271	0.319	0.283	0.302	0.240
4	0.334	0.287	0.331	0.220	0.367	0.277	0.388	0.276
6	0.371	0.329	0.354	0.219	0.309	0.257	0.301	0.231
8	0.286	0.221	0.296	0.223	0.351	0.274	0.403	0.240
10	0.385	0.273	0.401	0.280	0.384	0.320	0.369	0.300
20	0.488	0.341	0.585	0.316	0.586	0.416	0.749	0.390
30	0.418	0.340	0.433	0.295	0.654	0.476	0.907	0.485
40	0.521	0.434	0.658	0.263	0.686	0.497	0.951	0.542
50	0.524	0.377	0.617	0.385	0.712	0.538	0.963	0.584
100	0.650	0.460	0.904	0.487	0.771	0.652	0.972	0.673
	$FT_{RoBERTa_B}$				$FT_{RoBERTa_L}$			
Shots	Acc	$F1_C$	$F1_N$	$F1_S$	Acc	$F1_C$	$F1_N$	$F1_S$
2	0.333	0.145	0.209	0.157	0.334	0.137	0.249	0.167
4	0.335	0.215	0.187	0.157	0.333	0.139	0.157	0.250
6	0.334	0.189	0.175	0.172	0.333	0.132	0.173	0.216
8	0.333	0.169	0.214	0.140	0.334	0.145	0.194	0.202
10	0.333	0.153	0.175	0.197	0.334	0.171	0.175	0.212
20	0.340	0.195	0.185	0.207	0.428	0.383	0.439	0.275
30	0.359	0.223	0.239	0.196	0.500	0.383	0.640	0.416
40	0.384	0.290	0.266	0.260	0.564	0.418	0.759	0.454
50	0.405	0.391	0.303	0.291	0.587	0.472	0.794	0.450
100	0.557	0.441	0.727	0.443	0.674	0.544	0.882	0.565

Table 5: Few-Shot FT Performance on FEVER Three-way Veracity Classification. Acc stands for accuracy; $F1_C$, $F1_N$ and $F1_S$ stands for F1 score for "Contradict", "Neutral" and "Support" respectively.

C Detailed Standard Deviation Scores

Here we report detailed standard deviation scores of the three conducted experiments over multiple runs.

C.1 FEVER Binary Veracity Classification

Table 9 reports detailed few-shot performance for PB_{BERT_B} and PB_{BERT_L}.

Table 10 table reports detailed few-shot performance for FT_{BERT_B}, FT_{BERT_L}, $FT_{RoBERTa_B}$

	$SEED_{BERT_B}$				$SEED_{BERT_L}$			
Shots	Acc	$F1_C$	$F1_N$	$F1_S$	Acc	$F1_C$	$F1_N$	$F1_S$
2	0.383	0.331	0.216	0.431	0.392	0.290	0.252	0.461
4	0.459	0.360	0.501	0.476	0.468	0.336	0.522	0.489
6	0.519	0.389	0.664	0.514	0.513	0.385	0.653	0.493
8	0.546	0.417	0.726	0.510	0.536	0.397	0.706	0.485
10	0.559	0.424	0.744	0.519	0.554	0.368	0.731	0.535
20	0.594	0.445	0.805	0.528	0.580	0.413	0.768	0.528
30	0.604	0.455	0.817	0.527	0.579	0.412	0.774	0.511
40	0.617	0.464	0.821	0.549	0.589	0.420	0.776	0.532
50	0.622	0.469	0.819	0.556	0.590	0.418	0.776	0.537
100	0.635	0.477	0.821	0.585	0.594	0.422	0.774	0.551

	$SEED_{BERT_{NLI}}$			
Shots	Acc	$F1_C$	$F1_N$	$F1_S$
2	0.463	0.371	0.226	0.599
4	0.534	0.431	0.462	0.625
6	0.586	0.476	0.592	0.656
8	0.595	0.490	0.625	0.651
10	0.612	0.489	0.661	0.664
20	0.653	0.532	0.721	0.684
30	0.670	0.550	0.744	0.690
40	0.681	0.559	0.756	0.699
50	0.687	0.560	0.758	0.711
100	0.694	0.567	0.767	0.714

Table 6: Few-Shot SEED Performance on FEVER Three-way Veracity Classification. Acc stands for accuracy; $F1_C$, $F1_N$ and $F1_S$ stands for F1 score for "Contradict", "Neutral" and "Support" respectively.

and $FT_{RoBERTa_L}$.

Table 11 reports detailed few-shot performance for $SEED_{BERT_B}$, $SEED_{BERT_L}$ and $SEED_{BERT_{NLI}}$.

C.2 FEVER Three-way Veracity Classification

Table 12 reports detailed few-shot performance for FT_{BERT_B}, FT_{BERT_L}, $FT_{RoBERTa_B}$ and $FT_{RoBERTa_L}$. Table 13 reports detailed few-shot performance for $SEED_{BERT_B}$, $SEED_{BERT_L}$ and $SEED_{BERT_{NLI}}$.

C.3 SCIFACT Three-way Veracity Classification

Table 14 reports detailed few-shot performance for FT_{BERT_B}, FT_{BERT_L}, $FT_{RoBERTa_B}$ and $FT_{RoBERTa_L}$. Table 15 reports detailed few-shot performance for $SEED_{BERT_B}$, $SEED_{BERT_L}$ and $SEED_{BERT_{NLI}}$.

	FT_{BERT_B}				FT_{BERT_L}			
Shots	Acc	$F1_C$	$F1_N$	$F1_S$	Acc	$F1_C$	$F1_N$	$F1_S$
2	0.326	0.111	0.249	0.200	0.328	0.154	0.237	0.179
4	0.341	0.238	0.222	0.160	0.333	0.175	0.238	0.191
6	0.334	0.180	0.245	0.157	0.340	0.155	0.180	0.252
8	0.333	0.149	0.233	0.214	0.335	0.222	0.203	0.165
10	0.328	0.143	0.254	0.178	0.340	0.259	0.225	0.121
20	0.381	0.210	0.414	0.184	0.416	0.310	0.434	0.313
30	0.415	0.230	0.516	0.232	0.479	0.353	0.573	0.396
40	0.417	0.257	0.541	0.247	0.510	0.387	0.649	0.417
50	0.458	0.323	0.588	0.291	0.531	0.404	0.675	0.480
100	0.519	0.414	0.686	0.424	0.558	0.486	0.720	0.478

	$FT_{RoBERTa_B}$				$FT_{RoBERTa_L}$			
Shots	Acc	$F1_C$	$F1_N$	$F1_S$	Acc	$F1_C$	$F1_N$	$F1_S$
2	0.334	0.125	0.268	0.152	0.333	0.202	0.171	0.182
4	0.330	0.146	0.208	0.170	0.334	0.162	0.217	0.160
6	0.331	0.111	0.182	0.224	0.335	0.178	0.155	0.218
8	0.335	0.218	0.152	0.157	0.333	0.194	0.169	0.172
10	0.333	0.235	0.148	0.147	0.334	0.133	0.214	0.213
20	0.341	0.172	0.212	0.161	0.381	0.338	0.315	0.303
30	0.355	0.270	0.177	0.234	0.423	0.389	0.494	0.299
40	0.377	0.335	0.250	0.267	0.445	0.393	0.520	0.374
50	0.386	0.351	0.248	0.337	0.482	0.390	0.576	0.444
100	0.504	0.457	0.611	0.414	0.563	0.462	0.705	0.495

Table 7: Few-Shot FT Performance on SCIFACT Three-way Veracity Classification. Acc stands for accuracy; $F1_C$, $F1_N$ and $F1_S$ stands for F1 score for "Contradict", "Neutral" and "Support" respectively.

	$SEED_{BERT_B}$				$SEED_{BERT_L}$			
Shots	Acc	$F1_C$	$F1_N$	$F1_S$	Acc	$F1_C$	$F1_N$	$F1_S$
2	0.368	0.238	0.302	0.416	0.376	0.227	0.347	0.428
4	0.387	0.323	0.420	0.370	0.368	0.290	0.359	0.381
6	0.400	0.302	0.454	0.394	0.395	0.302	0.419	0.413
8	0.415	0.325	0.483	0.405	0.413	0.352	0.437	0.400
10	0.421	0.326	0.496	0.422	0.425	0.326	0.470	0.430
20	0.443	0.334	0.554	0.432	0.445	0.327	0.533	0.447
30	0.441	0.308	0.566	0.431	0.454	0.303	0.575	0.447
40	0.450	0.301	0.584	0.438	0.461	0.296	0.587	0.466
50	0.453	0.302	0.591	0.441	0.464	0.288	0.599	0.466
100	0.451	0.286	0.594	0.438	0.464	0.274	0.607	0.464

	$SEED_{BERT_{NLI}}$			
Shots	Acc	$F1_C$	$F1_N$	$F1_S$
2	0.399	0.341	0.277	0.451
4	0.458	0.399	0.368	0.530
6	0.472	0.431	0.393	0.531
8	0.480	0.416	0.441	0.538
10	0.500	0.432	0.456	0.578
20	0.509	0.424	0.494	0.580
30	0.517	0.439	0.505	0.582
40	0.521	0.451	0.515	0.577
50	0.527	0.453	0.526	0.581
100	0.533	0.468	0.531	0.582

Table 8: Few-Shot SEED Performance on SCIFACT Three-way Veracity Classification. Acc stands for accuracy; $F1_C$, $F1_N$ and $F1_S$ stands for F1 score for "Contradict", "Neutral" and "Support" respectively.

	PB_{BERT_B}			PB_{BERT_L}		
Shots	Acc	$F1_S$	$F1_{Not}$	Acc	$F1_S$	$F1_{Not}$
2	0.019	0.054	0.149	0.020	0.053	0.144
4	0.014	0.042	0.099	0.010	0.043	0.074
6	0.015	0.051	0.105	0.014	0.046	0.102
8	0.015	0.047	0.108	0.015	0.048	0.107
10	0.015	0.046	0.107	0.013	0.049	0.094
20	0.011	0.090	0.116	0.020	0.102	0.160
30	0.008	0.095	0.072	0.005	0.063	0.047
40	0.008	0.059	0.069	0.005	0.038	0.038
50	0.008	0.064	0.071	0.004	0.049	0.047
100	0.007	0.047	0.056	0.004	0.028	0.033

Table 9: Few-Shot PB Standard Deviation on FEVER Binary Veracity Classification. Acc stands for accuracy; $F1_S$, $F1_{Not}$ stands for F1 score for "Support" and "Not_Support" respectively.

	FT_{BERT_B}			FT_{BERT_L}		
Shots	Acc	$F1_S$	$F1_{Not}$	Acc	$F1_S$	$F1_{Not}$
2	0.025	0.326	0.273	0.038	0.313	0.255
4	0.025	0.322	0.285	0.039	0.310	0.248
6	0.035	0.320	0.271	0.043	0.302	0.245
8	0.039	0.320	0.262	0.039	0.310	0.252
10	0.034	0.316	0.260	0.033	0.316	0.267
20	0.056	0.305	0.174	0.066	0.201	0.176
30	0.053	0.307	0.186	0.072	0.191	0.084
40	0.063	0.300	0.111	0.063	0.116	0.156
50	0.070	0.262	0.075	0.054	0.047	0.138
100	0.074	0.197	0.078	0.037	0.041	0.064
	$FT_{RoBERTa_B}$			$FT_{RoBERTa_L}$		
Shots	Acc	$F1_S$	$F1_{Not}$	Acc	$F1_S$	$F1_{Not}$
2	0.003	0.320	0.326	0.003	0.319	0.317
4	0.005	0.319	0.308	0.003	0.323	0.320
6	0.002	0.326	0.325	0.005	0.309	0.319
8	0.005	0.300	0.312	0.001	0.330	0.331
10	0.002	0.320	0.325	0.006	0.319	0.326
20	0.008	0.313	0.306	0.022	0.233	0.227
30	0.009	0.310	0.303	0.018	0.231	0.232
40	0.013	0.285	0.268	0.028	0.165	0.146
50	0.022	0.257	0.269	0.027	0.096	0.115
100	0.032	0.164	0.166	0.063	0.126	0.133

Table 10: Few-Shot FT Standard Deviation on FEVER Binary Veracity Classification. Acc stands for accuracy; $F1_S$, $F1_{Not}$ stands for F1 score for "Support" and "Not_Support" respectively.

	$SEED_{BERT_B}$			$SEED_{BERT_L}$		
Shots	Acc	$F1_S$	$F1_{Not}$	Acc	$F1_S$	$F1_{Not}$
2	0.045	0.109	0.165	0.046	0.119	0.171
4	0.044	0.172	0.125	0.062	0.118	0.060
6	0.046	0.130	0.082	0.060	0.088	0.071
8	0.063	0.164	0.078	0.051	0.076	0.068
10	0.055	0.121	0.056	0.054	0.108	0.069
20	0.022	0.025	0.051	0.016	0.022	0.044
30	0.014	0.020	0.030	0.023	0.037	0.035
40	0.007	0.008	0.022	0.013	0.020	0.027
50	0.006	0.009	0.020	0.011	0.011	0.027
100	0.003	0.005	0.011	0.006	0.011	0.012
	$SEED_{BERT_{NLI}}$					
Shots	Acc		$F1_S$		$F1_{Not}$	
2	0.095		0.125		0.137	
4	0.115		0.138		0.107	
6	0.045		0.058		0.051	
8	0.078		0.096		0.073	
10	0.081		0.102		0.077	
20	0.011		0.026		0.019	
30	0.015		0.022		0.018	
40	0.013		0.019		0.012	
50	0.009		0.015		0.009	
100	0.006		0.011		0.007	

Table 11: Few-Shot SEED Standard Deviation on FEVER Binary Veracity Classification. Acc stands for accuracy; $F1_S$, $F1_{Not}$ stands for F1 score for "Support" and "Not_Support" respectively.

	FT_{BERT_B}				FT_{BERT_L}			
Shots	Acc	$F1_C$	$F1_N$	$F1_S$	Acc	$F1_C$	$F1_N$	$F1_S$
2	0.229	0.258	0.399	0.305	0.237	0.233	0.406	0.288
4	0.232	0.278	0.398	0.284	0.243	0.243	0.445	0.289
6	0.229	0.296	0.429	0.290	0.253	0.249	0.429	0.281
8	0.238	0.256	0.382	0.276	0.262	0.249	0.443	0.284
10	0.239	0.280	0.444	0.310	0.219	0.243	0.442	0.295
20	0.196	0.266	0.425	0.301	0.128	0.211	0.370	0.261
30	0.202	0.236	0.419	0.287	0.064	0.162	0.186	0.180
40	0.179	0.229	0.397	0.271	0.033	0.138	0.032	0.164
50	0.163	0.206	0.427	0.262	0.037	0.154	0.012	0.132
100	0.065	0.132	0.183	0.206	0.035	0.064	0.006	0.105
	$FT_{RoBERTa_B}$				$FT_{RoBERTa_L}$			
Shots	Acc	$F1_C$	$F1_N$	$F1_S$	Acc	$F1_C$	$F1_N$	$F1_S$
2	0.008	0.224	0.242	0.229	0.005	0.198	0.244	0.224
4	0.014	0.238	0.224	0.224	0.006	0.213	0.225	0.234
6	0.010	0.235	0.228	0.231	0.002	0.213	0.233	0.241
8	0.001	0.235	0.242	0.213	0.008	0.222	0.235	0.231
10	0.006	0.224	0.230	0.239	0.006	0.217	0.224	0.238
20	0.013	0.226	0.229	0.231	0.068	0.167	0.244	0.182
30	0.040	0.239	0.253	0.228	0.066	0.123	0.157	0.095
40	0.057	0.221	0.266	0.221	0.066	0.105	0.151	0.121
50	0.068	0.150	0.275	0.185	0.077	0.098	0.188	0.106
100	0.068	0.102	0.157	0.130	0.102	0.134	0.193	0.130

Table 12: Few-Shot FT Standard Deviation on FEVER Three-way Veracity Classification. Acc stands for accuracy; $F1_C$, $F1_N$ and $F1_S$ stands for F1 score for "Contradict", "Neutral" and "Support" respectively.

	$SEED_{BERT_B}$				$SEED_{BERT_L}$			
Shots	Acc	$F1_C$	$F1_N$	$F1_S$	Acc	$F1_C$	$F1_N$	$F1_S$
2	0.023	0.146	0.108	0.144	0.041	0.189	0.166	0.115
4	0.016	0.119	0.080	0.113	0.042	0.154	0.135	0.128
6	0.026	0.080	0.070	0.077	0.031	0.116	0.063	0.104
8	0.030	0.066	0.063	0.065	0.034	0.114	0.063	0.108
10	0.029	0.062	0.069	0.042	0.018	0.099	0.041	0.069
20	0.020	0.054	0.015	0.035	0.011	0.069	0.010	0.076
30	0.018	0.043	0.015	0.044	0.005	0.073	0.009	0.089
40	0.017	0.040	0.015	0.038	0.006	0.062	0.008	0.071
50	0.013	0.042	0.012	0.038	0.008	0.060	0.010	0.069
100	0.016	0.033	0.010	0.032	0.011	0.058	0.006	0.049

	$SEED_{BERT_{NLI}}$			
Shots	Acc	$F1_C$	$F1_N$	$F1_S$
2	0.055	0.110	0.176	0.027
4	0.051	0.083	0.142	0.086
6	0.040	0.063	0.088	0.050
8	0.043	0.046	0.075	0.062
10	0.036	0.048	0.060	0.032
20	0.013	0.026	0.026	0.031
30	0.024	0.016	0.025	0.052
40	0.020	0.013	0.019	0.037
50	0.019	0.024	0.019	0.029
100	0.011	0.013	0.010	0.024

Table 13: Few-Shot SEED Standard Deviation on FEVER Three-way Veracity Classification. Acc stands for accuracy; $F1_C$, $F1_N$ and $F1_S$ stands for F1 score for "Contradict", "Neutral" and "Support" respectively.

	FT_{BERT_B}				FT_{BERT_L}			
Shots	Acc	$F1_C$	$F1_N$	$F1_S$	Acc	$F1_C$	$F1_N$	$F1_S$
2	0.034	0.203	0.185	0.244	0.037	0.229	0.178	0.240
4	0.039	0.248	0.192	0.226	0.049	0.244	0.174	0.235
6	0.035	0.238	0.194	0.230	0.032	0.236	0.183	0.249
8	0.050	0.226	0.186	0.247	0.042	0.251	0.173	0.231
10	0.041	0.220	0.175	0.240	0.037	0.256	0.184	0.208
20	0.054	0.246	0.144	0.244	0.064	0.202	0.208	0.216
30	0.072	0.231	0.108	0.257	0.067	0.157	0.176	0.185
40	0.072	0.201	0.088	0.233	0.054	0.144	0.123	0.158
50	0.068	0.173	0.109	0.247	0.048	0.118	0.107	0.115
100	0.043	0.085	0.063	0.146	0.044	0.064	0.078	0.082

	$FT_{RoBERTa_B}$				$FT_{RoBERTa_L}$			
Shots	Acc	$F1_C$	$F1_N$	$F1_S$	Acc	$F1_C$	$F1_N$	$F1_S$
2	0.015	0.205	0.242	0.221	0.017	0.230	0.229	0.228
4	0.013	0.219	0.237	0.232	0.016	0.223	0.238	0.229
6	0.011	0.202	0.233	0.243	0.011	0.231	0.220	0.238
8	0.017	0.242	0.229	0.227	0.012	0.237	0.230	0.225
10	0.010	0.237	0.223	0.219	0.019	0.204	0.232	0.240
20	0.027	0.233	0.250	0.233	0.040	0.142	0.214	0.179
30	0.032	0.219	0.234	0.222	0.043	0.105	0.112	0.140
40	0.043	0.196	0.237	0.206	0.053	0.102	0.135	0.098
50	0.040	0.147	0.238	0.170	0.045	0.101	0.130	0.072
100	0.038	0.058	0.084	0.084	0.067	0.116	0.102	0.102

Table 14: Few-Shot FT Standard Deviation on SCI-FACT Three-way Veracity Classification. Acc stands for accuracy; $F1_C$, $F1_N$ and $F1_S$ stands for F1 score for "Contradict", "Neutral" and "Support" respectively.

	$SEED_{BERT_B}$				$SEED_{BERT_L}$			
Shots	Acc	$F1_C$	$F1_N$	$F1_S$	Acc	$F1_C$	$F1_N$	$F1_S$
2	0.025	0.134	0.130	0.114	0.045	0.116	0.131	0.121
4	0.044	0.075	0.066	0.107	0.042	0.082	0.084	0.124
6	0.037	0.087	0.073	0.112	0.036	0.090	0.082	0.095
8	0.030	0.086	0.048	0.103	0.027	0.098	0.066	0.112
10	0.033	0.071	0.086	0.070	0.036	0.130	0.084	0.072
20	0.032	0.060	0.045	0.037	0.030	0.080	0.053	0.051
30	0.025	0.042	0.032	0.026	0.038	0.078	0.058	0.063
40	0.023	0.017	0.036	0.027	0.030	0.063	0.048	0.038
50	0.019	0.023	0.033	0.021	0.023	0.055	0.028	0.034
100	0.015	0.022	0.029	0.020	0.023	0.029	0.037	0.036

	$SEED_{BERT_{NLI}}$			
Shots	Acc	$F1_C$	$F1_N$	$F1_S$
2	0.062	0.055	0.077	0.174
4	0.053	0.048	0.090	0.130
6	0.052	0.052	0.085	0.141
8	0.055	0.058	0.082	0.121
10	0.032	0.039	0.073	0.040
20	0.028	0.058	0.028	0.030
30	0.022	0.050	0.043	0.020
40	0.026	0.051	0.037	0.025
50	0.032	0.054	0.046	0.027
100	0.018	0.026	0.031	0.027

Table 15: Few-Shot SEED Standard Deviation on SCI-FACT Three-way Veracity Classification. Acc stands for accuracy; $F1_C$, $F1_N$ and $F1_S$ stands for F1 score for "Contradict", "Neutral" and "Support" respectively.