Few-shot Claim Verification for Automated Fact Checking

Xia Zeng

Submitted in partial fulfilment of the requirement for the degree of *Doctor of Philosophy*

School of Electronic Engineering and Computer Science

Queen Mary University of London

16 June 2024

Few-shot Claim Verification for Automated Fact Checking

Xia Zeng

Abstract

In an era characterized by the rapid expansion of online information and the widespread dissemination of misinformation, automated fact-checking has emerged as an essential area of research. As digital platforms continue to proliferate, the necessity for accurate and efficient fact-checking mechanisms is attracting increasing interest. Automated fact-checking systems address two main tasks: claim detection and claim validation. Claim detection involves identifying sentences or text snippets containing assertions or claims potentially subject to fact-checking. Claim validation, a multifaceted endeavor, encompasses evidence retrieval and claim verification. During evidence retrieval, relevant information or evidence that may support or refute a given claim is obtained. Claim verification, on the other hand, entails assessing the veracity of a claim by comparing it against available evidence. Typically framed as a natural language inference (NLI) problem, claim verification requires the model to determine whether a claim is supported, refuted, or there is not enough information to reach a verdict.

In this thesis, we explore challenges inherent in claim verification, with a focus on few-shot scenarios where limited labeled data and computational resources pose significant constraints. We introduce three innovative methods tailored to tackle these challenges: Semantic Embedding Element-wise Difference (SEED), Micro Analysis of Pairwise Language Evolution (MAPLE), and Active learning with Pattern Exploiting Training models (Active PETs). SEED, a novel vector-based approach, leverages semantic differences in claim-evidence pairs to perform claim verification in few-shot scenarios. By creating class representative vectors, SEED enables efficient claim verification even with limited training data. Comparative evaluations against previous state-of-the-art methods demonstrate SEED's consistent improvements in few-shot settings. MAPLE is another pioneering approach to few-shot claim verification, harnessing a small seq2seq model and a novel semantic measure to explore the alignment between claims and evidence. Utilizing micro analysis of pairwise language evolution, MAPLE achieves significant performance improvements over state-of-the-art baselines across multiple automated fact-checking datasets. Active PETs presents a novel ensemble-based active learning approach for data annotation prioritization in few-shot claim verification. By utilizing an ensemble of Pattern Exploiting Training (PET) models based on various pre-trained language models, Active PETs effectively selects unlabelled data for annotation, consistently outperforming baseline active learning methods. Its integrated oversampling strategy further enhances performance, demonstrating the potential of active learning techniques in optimizing claim verification workflows.

Together, these methods represent significant advancements in claim verification research, offering scalable and practical solutions. Through extensive experimentation and comparative analysis, this thesis evaluates the effectiveness of each method on various dataset configurations and provides valuable insights into their strengths and weaknesses. Furthermore, by identifying potential extensions and areas for refinement, the thesis lays the groundwork for future research endeavors in this critical field of artificial intelligence.

Declaration

I hereby declare that this thesis has been composed by myself and that it describes my own work. The only collaboration work presented is the section B.1 Claim Detection in Appendix B Related Tasks with Amani S. Abumansour. The thesis has not been submitted, either in the same or different form, to this or any other university for a degree. All verbatim extracts are distinguished by quotation marks, and all sources of information have been acknowledged.

Acknowledgements

I express my sincere gratitude to the following individuals and institutions who have played a pivotal role in the completion of this research:

I am grateful to QMUL-CSC for providing funding support and Queen Mary's Apocrita HPC facility, backed by QMUL Research-IT, for their essential computational resources. I extend my appreciation to the Allen Turing Institute and HSBC-UK for the enriching internship opportunity.

Special thanks to my dedicated supervisor, Dr. Arkaitz Zubiaga, for his unwavering support. His insightful conversations, swift feedback, and empathetic guidance have been invaluable, providing not only academic assistance but also emotional support during challenging times.

I acknowledge the collaborative efforts of my colleague Amani S. Abumansour in the surveying project on claim detection. I am indebted to Ji-Ung Lee for his valuable insights and comments on active learning. Prof. Maria Liakata and Prof. Massimo Poesio deserve thanks for their thoughtful comments and feedback on my reports. I appreciate the engaging conversations with Dr. Julian Hough, Christopher Schröder, and Dr. Yuerong Zhang. Additionally, I am thankful to the anonymous reviewers for their constructive comments.

A special acknowledgment is reserved for my husband, Dr. Chris Ahart, for his epic encouragement, comforting hugs, and steadfast support throughout this lengthy journey.

I express my heartfelt thanks to my family—Yongping Zeng, Youhong Chen, Ying Zeng, and Jinlan Yang—for keeping me loved and grounded. I am grateful to my friends—Wenkai Tay, Aiqi Jiang, Wenjie Yin, Tom Kaplan, Aida Halitaj, and Fatima Althani (in chronological order)—for keeping me socialized and providing a sense of community.

Finally, a big thank you to coffee and noodles, my loyal companions throughout this endeavor, for keeping me watered and fed.

Contents

List of Figures

List of Tables

Glossary

Active PETs Active Pattern Exploiting Trainings. Active PETs is a method for actively selecting unlabeled samples with an ensemble of PET models.

AL Active Learning. AL is a machine learning paradigm where the model proactively selects the subset of examples to be labeled next from the pool of unlabeled data.

ALPS Active Learning by Processing Surprisal. ALPS is a method that uses surprisal scores to select instances for annotation in active learning.

BADGE Batch Active learning by Diverse Gradient Embeddings. BADGE is an active learning method that batch selects diverse instances for annotation.

BERT Bidirectional Encoder Representations from Transformers. BERT is a transformer-based language model developed by Google for natural language processing tasks.

BioSentVec Biomedical Sentence Vectors. BioSentVec refers to vectors representing biomedical sentences, often used in natural language processing tasks in the biomedical domain.

BLEU Bilingual Evaluation Understudy. BLEU is a metric for evaluating the quality of machine-translated text based on n-gram overlap with reference translations.

BLEURT Bilingual Evaluation Understudy with Representations from Transformers. BLEURT is a metric for evaluating machine translation that leverages BERT-based representations.

CAL Contrastive Active Learning. CAL is an active learning approach that selects instances that are similar in the model feature space and yet the model outputs maximally different predictive likelihoods.

cFEVER Climate Fact Extraction and VERification. cFEVER is a variant of the FEVER dataset focused on climate change facts.

CLEF Conference and Labs of the Evaluation Forum. CLEF is an forum that established a framework of systematic evaluation of information access systems, primarily through experimentation on shared tasks.

DeBERTa Decoding-enhanced BERT with disentangled attention. DeBERTa is a variant of BERT with improved decoding and attention mechanisms.

DistilBERT Distilled BERT. DistilBERT is a smaller, faster version of the BERT model.

DistilRoBERTa Distilled RoBERTa. DistilRoBERTa is a smaller, faster version of the RoBERTa model.

ELECTRA Efficiently Learning an Encoder that Classifies Token Replacements Accurately. ELECTRA is a model that learns to discriminate between real and generated tokens.

FEVER Fact Extraction and VERification. FEVER is a dataset for evaluating fact extraction and verification systems.

FT/SFT Fine-Tuning / Supervised Fine-Tuning. FT/SFT is the process of further training a pre-trained language model on a specific task or dataset.

GPT Generative Pre-trained Transformer. GPT is a type of transformer-based language model developed by OpenAI for natural language processing tasks.

IR Information Retrieval. IR is the process of obtaining information from a collection of documents.

KILT Knowledge Intensive Language Tasks. KILT is a benchmark for evaluating language models on knowledge-intensive tasks.

LLaMA 2 Large Language Model Meta AI 2. LLaMA 2 is a recent generative large language model with multi-billion parameters from Meta AI that uses an optimized transformer architecture.

LLM Large Language Model. LLM is a language model that is larger and more powerful than traditional models.

LoRA Low-Rank Adaptation. LoRA is an efficient method for adapting language models to new tasks with low-rank matrices.

MAPLE Micro Analysis of Pairwise Language Evolution. MAPLE is a few-shot claim verification method that utilises signals from the language transition process during seq2seq training.

METEOR Metric for Evaluation of Translation with Explicit ORdering. METEOR is a metric for evaluating the quality of machine translation based on the weighted harmonic F1 of unigram precision and recall.

MLM Masked Language Modeling. MLM is a task where a model is trained to predict masked tokens in a sentence, used in pre-training language models.

MNLI Multi-Genre Natural Language Inference. MNLI is a dataset for evaluating natural language inference systems across multiple genres of text.

MTTR Maas Type-Token Ratio. MTTR is a variant of the type-token ratio used in natural language processing.

NLG Natural Language Generation. NLG is the task of generating natural language text from text or other forms of input.

NLI Natural Language Inference. NLI is a task in natural language processing where the goal is to determine the logical relationship between two pieces of text.

NLP Natural Language Processing. NLP is a field of artificial intelligence focused on the interaction between computers and human languages.

NLPO Natural Language Policy Optimization. NLPO is a framework for training natural language processing models using reinforcement learning.

PB Perplexity-Based. PB is a method that uses perplexity scores to perform few-shot claim verification.

PET Pattern Exploiting Training. PET is a method for training language models with verbalizers and patterns.

PLM Pre-trained Language Model. PLM is a language model that has been pre-trained on a large corpus of text.

QBC Query-By-Committee. QBC is an active learning strategy where a committee of models is used to select instances for annotation.

RL Reinforcement Learning. RL is a machine learning paradigm where an agent learns to make decisions by interacting with an environment and receiving rewards.

RLHF Reinforcement Learning from Human Feedback. RLHF is a framework where a reward model is first trained from human feedback and then used to optimize the performance of an agent through reinforcement learning.

RoBERTa Robustly optimized BERT approach. RoBERTa is a variant of BERT with improved training dynamics and performance.

ROUGE Recall-Oriented Understudy for Gisting Evaluation. ROUGE is a set of metrics for evaluating automatic summarization and machine translation.

RTE Recognizing Textual Entailment. RTE is a task in natural language processing where the goal is to determine if one piece of text entails another.

SEED Semantic Embedding Element-wise Difference. SEED is a vector-based method for few-shot claim verification.

SemSim Semantic Similarity. SemSim refers to a method that calculates the degree to which two pieces of text convey the same meaning using sentence representations from language models.

SOTA State Of The Art. SOTA refers to the current best performance on a particular task or problem.

Chapter 1

Introduction

1.1 Introduction

With the increased availability of information through online platforms, people increasingly use the Web to access up-to-date information and to learn about the latest news and events. Along with the increased availability of information, this has also led to an increase of misinformation, which can lead society to making wrong decisions in your life due to the inaccuracy of the information. As such, identifying when online information is inaccurate becomes crucial as a means to support people to be aware of misinformation. 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 deals with identifying the set of sentences, out of a long text, deemed capable of being fact-checked (Konstantinovskiy et al., 2021), and (2) claim validation, which assess the veracity of a claim by checking it against a piece of evidence, with a two-step process that does evidence retrieval first, followed by claim verification for claims (Pradeep et al., 2021). The evidence retrieval component obtains, typically from a database, the most relevant piece(s) of evidence for a given claim. Once the evidence is retrieved, the claim verification component determines the level of support the evidence gives to the claim.

As a fertile research area where numerous methods have been proposed and tested to support automated fact-checking, substantial improvements have been achieved in the performance of claim validation models when a considerable amount of training data is available (Pradeep et al.,

2021; Li et al., 2021; Zeng and Zubiaga, 2021; Zhang et al., 2021; Wadden et al., 2022). As a key component of the claim validation pipeline, the claim verification[1] 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 (Thorne et al., 2018a; Wadden et al., 2020; Lee et al., 2021). It is predominantly tackled as a natural language inference (NLI) task: given a claim c and a piece of evidence e, predict the veracity label for the claim c which can be one of 'SUPPORTS', 'REFUTES', and 'NOT_ENOUGH_INFO'. The FEVER (Thorne et al., 2018a) dataset presents the following 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." Adding to these, the Climate FEVER (Diggelmann et al., 2021) dataset offers an illustrative example where the claim "Coral atolls grow as sea levels rise." is supported by the evidence "Gradual sea-level rise also allows for coral polyp activity to raise the atolls with the sea level,". Similarly, in the SciFact (Wadden et al., 2020) dataset, the claim "Fz/PCP-dependent Pk localizes to the anterior membrane of notochord cells during zebrafish neurulation." receives a 'NOT_ENOUGH_INFO' label when paired with evidence "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."

While the majority of previous work tackles the problem with fully supervised methods where there is a good amount of labeled training data available (Li et al., 2021; Zeng and Zubiaga, 2021; Zhang et al., 2021; Wadden et al., 2022; Rana et al., 2022b,a), deploying these methods face practicality issues. Emerging domains of misinformation often involve novel claims, limiting the availability of relevant labeled data. This is the case, for example, of claims associated with newly emerging topics such as COVID-19, which at the time of becoming widely discussed in society lacked sufficient data, not least instances which were labeled by fact-checkers. Indeed, the claims needing fact-checking can be diverse, ranging from political claims to health related claims, including other subjects such as finance and more general news. This leads to a diversity of datasets used in automated fact-checking, sometimes including more general-domain datasets, in other cases including domain-specific datasets; in this thesis, we are interested in investigating the impact of this diversity in the developing of automated fact-checking systems.

[1]The task is sometimes referred to as veracity classification (Lee et al., 2021).

In addition, fact-checkers often need to evaluate claims with time constraints, limiting the time allowed for conducting extensive fine-tuning of pretrained language models (PLMs). Hence, performing claim verification in few-shot scenarios, where a model has seen very limited labeled data that resembles what will be seen during the test phase, is of particular importance in the real-world combat of misinformation.

Interestingly, the availability of unlabelled data can often be abundant in the context of automated fact-checking, but given the cost and effort of labeling this data, one needs to be selective in labeling a small subset. In these circumstances, rather than randomly sampling this subset, in this thesis we hypothesize that we can optimize the selection of candidate instances to be labeled through active learning, and that we can strategically design models to make better use of the limited labeling budget than existing approaches, such that it leads to overall improved few-shot performance.

1.2 Research Questions and Objectives

The overarching aim of this thesis is to study the extensibility of automated fact-checking models to the scenario with limited training data, i.e. in few-shot settings. This involves both identifying weaknesses of existing, state-of-the-art models, as well as furthering their ability by proposing new and improved approaches, in turn evaluating their effectiveness with increasing levels of challenge where the number of labeled samples decreases.

To address this aim, we address the following more specific research questions:

- **RQ1:** How do the challenges vary between different types of datasets, including domain-specific versus more general, and synthetic versus non-synthetic data, as well as different dataset configurations such as oracle versus retrieved evidence configurations?
- **RQ2:** What are the existing and novel few-shot claim verification methods, and how do they tackle the obstacles presented by scarce annotations, tight annotation budgets, and the limitations imposed by restricted computing resources?
- **RQ3:** What are the comparative strengths and weaknesses of various few-shot claim verification methods, and which method is most suitable for specific scenarios?

To tackle these questions, our objective is to devise novel few-shot claim verification methods that exhibit strong performance across diverse domains, demonstrate robustness to noisy evidence,

and offer scalability and practicality in implementation.

1.3 Contributions to the Field

This research significantly contributes to the active research area of automated fact-checking by studying the challenges these systems face and how these can be improved in few-shot settings. By introducing novel questions and solutions, and providing insights into the challenges and opportunities in the field, the contributions of this thesis aim to advance the understanding and capabilities of claim verification systems. This thesis makes the first comprehensive contribution to claim verification in few-shot settings, which had been understudied.

Contributions Our contributions in this study are outlined as follows:

- We conducted extensive experiments to explore the challenges in few-shot claim verification across multiple datasets and various dataset configurations, providing insights into the complexities of the task.
- We adapted established few-shot Natural Language Processing (NLP) methods such as PET and LLaMA 2 into the domain of few-shot claim verification, showcasing their applicability and effectiveness in addressing verification challenges.
- We introduced two novel few-shot claim verification methods, SEED and MAPLE, designed specifically to overcome the hurdles posed by limited annotations and computing resources, contributing innovative solutions to the field.
- We proposed a novel paradigm called Active PETs, which combines active learning with few-shot claim verification to optimize data annotation prioritization, thereby maximizing the utility of limited annotation budgets for improved outcomes.
- We conducted comprehensive analyses to evaluate the strengths and weaknesses of each explored method, offering recommendations on their respective use cases to guide practitioners in selecting the most suitable approach for their specific scenarios.

1.4 Publications

Research conducted as part of this PhD and reported in this thesis has been published in academic venues as follows:

- **Xia Zeng** and Arkaitz Zubiaga. 2024. MAPLE: Micro Analysis of Pairwise Language Evolution for Few-Shot Claim Verification. In Findings of the Association for Computational Linguistics: EACL 2024, pages 1177–1196, St. Julian's, Malta. Association for Computational Linguistics.

- **Xia Zeng** and Arkaitz Zubiaga. 2023. Active PETs: Active Data Annotation Prioritisation for Few-Shot Claim Verification with Pattern Exploiting Training. In Findings of the Association for Computational Linguistics: EACL 2023, pages 190–204, Dubrovnik, Croatia. Association for Computational Linguistics.

- **Xia Zeng** and Arkaitz Zubiaga. 2022. Aggregating pairwise semantic differences for few-shot claim verification. PeerJ Computer Science 8:e1137 https://doi.org/10.7717/peerj-cs.1137

- **Xia Zeng**, Amani S. Abumansour and Arkaitz Zubiaga. 2021. Automated fact-checking: A survey. Language and Linguistics Compass, e12438. https://doi.org/10.1111/lnc3.12438

- **Xia Zeng** and Arkaitz Zubiaga. 2021. QMUL-SDS at SCIVER: Step-by-step binary classification for scientific claim verification. In Proceedings of the Second Workshop on Scientific Document Processing (pp. 116–123). Association for Computational Linguistics.

1.5 Thesis Structure

The thesis is structured in seven chapters, as follows:

Chapter 1 provides introductory context for the growing importance and impact of automated fact-checking, highlighting the demand for improved few-shot claim verification methods. It discusses the motivation and contributions of this thesis, and sets the stage for the subsequent chapters.

Chapter 2 offers a comprehensive literature review within the realm of automated fact-checking, structured into four key sections. It encompasses the evolution and significance of automated fact-checking, delves into claim validation techniques, highlights the challenges and solutions in few-shot claim verification, and concludes with the methodological foundations central to our innovative solutions. This chapter aims to set a solid foundation for the thesis, situating our contributions within the broader context of NLP and AI advancements.

Chapter 3 provides an overview of the experimental framework for all the experiments presented within this thesis, detailing the shared resources utilized in our studies. This includes a comprehensive discussion on the datasets and baseline models employed, the formulation of the problem, and the evaluation metrics used. By consolidating these core components in a single chapter, we aim to provide a clear, unified foundation for the novel contributions presented in the subsequent chapters.

Chapter 4 introduces Semantic Embedding Element-wise Difference (SEED), a novel vector-based method designed for few-shot claim verification. SEED leverages pairwise semantic differences in claim-evidence pairs, simulating class representative vectors to enhance classification accuracy. Comparative evaluations against competitive baselines demonstrate consistent improvements in few-shot settings.

Chapter 5 introduces Micro Analysis of Pairwise Language Evolution (MAPLE), a pioneering approach for few-shot claim verification. MAPLE leverages a small seq2seq model and a novel semantic measure to explore the alignment between claims and evidences, demonstrating significant performance improvements over state-of-the-art baselines across multiple fact-checking datasets.

Chapter 6 proposes Active PETs, a novel weighted approach for active data annotation prioritization in few-shot claim verification. By utilizing an ensemble of Pattern Exploiting Training (PET) models based on various language models, Active PETs effectively selects unlabelled data for annotation, consistently outperforming baseline methods and achieving further improvements with an integrated oversampling strategy.

Chapter 7 summarizes the findings from the individual chapters to answer the proposed research questions, highlighting the contributions made and their implications for the field of automated fact-checking. It further identifies potential extensions of the proposed methodologies, areas for refinement, and new directions that can contribute to the ongoing evolution of claim verification techniques. This forward-looking perspective aims to inspire and guide future research endeavors in this dynamic and crucial field.

Additionally, this thesis is supplemented with four appendices. These appendices offer additional insights, detailed analyses, and supporting information, enriching the reader's understanding of the scope and impact of our work.

- **Appendix A** presents our participation in the QMUL@SCIVER shared task in the early

stages of this PhD, where we introduced a step-by-step binary classification approach to address the entire claim validation pipeline comprehensively. This appendix details the methodology behind our approach, illustrating how each step contributes to the overall objective of claim verification. Our experiments demonstrate that classification models can effectively complete the whole claim validation pipeline, showing potential broader impact for better claim verification models.

- **Appendix B** provides an extended literature review focused on related tasks within the domain of misinformation detection, such as claim detection and fake news detection, providing a broader context to our research.

- **Appendix C** includes additional comprehensive results to supplement the experiments reported in the main chapters. This appendix reinforces the robustness and reliability of our findings, and further validates our contributions to the field.

- **Appendix D** features a runtime report, which delves into the computational efficiency of our proposed methods. This section addresses practical considerations important for the implementation and deployment of automated fact-checking technologies, underscoring the feasibility of applying our research in real-world settings.

1.6 Summary

This introduction underscores the pivotal role of few-shot claim verification models in enhancing automated fact-checking systems, setting the stage for a detailed exploration of innovative, scalable, and practical methods for claim verification. Across seven chapters, this thesis embarks on a comprehensive journey from a broad literature review in automated fact-checking and few-shot claim verification, to a detailed discussion on shared experimental resources, before delving into the specific contributions of SEED, MAPLE, and Active PETs. These novel approaches collectively showcase the transformative potential of leveraging few-shot claim verification within natural language processing to address and mitigate misinformation effectively. Chapter 7 synthesizes the findings, evaluates the contributions, and explores future research directions, aimed at further advancing the field of automated fact-checking and beyond. Complementing the main content, four appendices provide additional insights, extending the discussion on related tasks, offering detailed additional experimental results, and providing runtime reports.

Chapter 2

Background

The domain of automated fact-checking stands at the forefront of addressing the proliferation of misinformation in the digital age. This background chapter delves into the intricate landscape of automated fact-checking, tracing its evolution from foundational concepts to the cutting-edge methodologies that define the field today. It begins by exploring the broad spectrum of automated fact-checking, laying the groundwork for understanding its critical role in contemporary information dissemination. The discussion then narrows to the specific challenges and innovations within claim validation, a core component that underscores the complexity of verifying information authenticity. Further, the chapter addresses the problem of few-shot claim verification, highlighting its significance in leveraging limited data to make substantial inferences—a critical capability in the fast-paced digital world where data scarcity is a common hurdle. We conclude with the methodological foundations that inform our approach, setting the stage for the novel contributions of this thesis. Through this focused literature review, we contextualize our work within the broader landscape of natural language processing and artificial intelligence.

2.1 Automated Fact-Checking

While online content continues to grow unprecedentedly, the spread of false information online increases the potential of misleading people and causing harm. This leads to an increasing demand on fact-checking, i.e. a task consisting in assessing the truthfulness of a claim (Vlachos and Riedel, 2014), where a claim is defined as ‘a factual statement that is under investigation’ (Hanselowski, 2020). As an indicator of the pressing need of fact-checking to support assessing the integrity of

information in circulation, a number of fact-checking organisations have been founded in recent years, e.g. FactCheck, PolitiFact, Full Fact, Snopes, Poynter and NewsGuard. At the time of this writing, the directory by the Duke Reporters' Lab[1] documents a total of 439 active fact-checking sites globally.

Fact-checkers continually conduct laborious manual fact-checking, which involves a complex set of tasks including: familiarising with the topic, identifying claims needing fact-checking, searching for evidence linked to a claim, checking source credibility, verifying the claim against the evidence collected and writing up an article that summarizes the assessment of a claim (Hanselowski, 2020). This however proves challenging, not least because the speed and efficiency of manual fact-checking cannot keep up with the pace at which online information is posted and circulated. The journalism community can benefit from tools that can support or, at least partially, automate the fact-checking process (Cohen et al., 2011; Hassan et al., 2017a; Thorne and Vlachos, 2018; Konstantinovskiy et al., 2021). This can be achieved primarily by automating more mechanical tasks, so that human effort can instead be dedicated to more knowledge-intensive tasks (Babakar and Moy, 2016). Restricting claims to those that are objectively fact-checkable makes the automation task more realistically achievable while reducing the volume of content needing manual fact-checking. Furthermore, recent progress in the fields of natural language processing (NLP), information retrieval (IR) and big data mining has demonstrated the potential for efficiently processing large-scale textual information online, which is also being leveraged in the context of automated fact-checking.

Researchers have developed valuable fact-checking datasets, pipelines and models, an effort which has also been supported by shared tasks, including HeroX fact checking challenge (Francis and Fact, 2016), Fake News Challenge (Pomerleau and Rao, 2017), ClaimBuster (Hassan et al., 2017b), RumourEval (Derczynski et al., 2017; Gorrell et al., 2018; Derczynski et al., 2017), FEVER (Thorne et al., 2018a, 2019; Aly et al., 2021), CLEF CheckThat! (Nakov et al., 2018; Elsayed et al., 2019; Barrón-Cedeño et al., 2020; Nakov et al., 2021b, 2022; Barrón-Cedeño et al., 2023), SCIVER (Wadden et al., 2020), and SEM-TAB-FACTS (Wang et al., 2021).

With different major concerns, proposed pipelines take various forms. For instance, ClaimBuster (Hassan et al., 2017b) designed a comprehensive pipeline of four components to verify web documents: a claim monitor that performs document retrieval; a claim spotter that performs

[1] `https://reporterslab.org/fact-checking/`

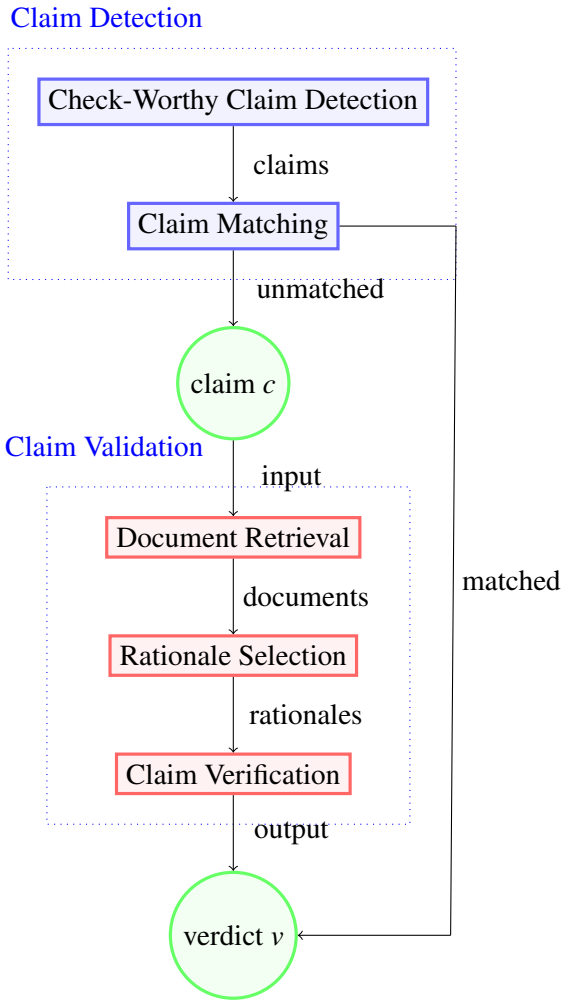

Figure 2.1: An Overview of Automated Fact-checking System.

claim detection; a claim matcher that matches a detected claim to fact-checked claims; a claim checker that performs evidence extraction and claim validation. A similar pipeline was proposed by CLEF CheckThat! (Nakov et al., 2021b), which in its 2021 edition included three subtasks: first, perform claim detection to detect claims that are check-worthy; second, determine whether a claim has been previously fact-checked; and third, perform claim validation to determine the factuality of the detected claims. In subsequent years, the CheckThat! shared task has continued innovating by adding more tasks on related challenges, including detection of previously fact-checked claims (Nakov et al., 2022), as well as multimodal fact-checking and detecting subjectivity in news articles (Alam et al., 2023). While some pipelines include claim detection, some are only designed to tackle claim validation, e.g. FEVER (Thorne et al., 2018a, 2019) and SCIVER (Wadden et al., 2020),[2] assuming check-worthy claims are already at hand. Figure 2.1 depicts a comprehensive fact-checking pipeline as discussed in this chapter and consisting of two components: (1) a claim detection component, which looks for claims that need checking and tries to find matches between claims when they are related to the same fact-check, and (2) a claim validation component, which retrieves the documents and rationales that can serve as evidence to fact-check a claim and ultimately performs the verification task, producing a verdict. In this thesis, we focus on claim validation due to a special interest in predicting the veracity of a given claim. Please see detailed overview on claim detection and other related tasks in Appendix B.

2.2 Claim Validation

As a component of the automated fact-checking pipeline, claim validation is formulated as 'the assignment of a truth value to a claim made in a particular context' (Vlachos and Riedel, 2014).

2.2.1 Approaches

In order to fulfill the task of claim validation, two different major approaches to verification have emerged: 1) the claim is verified against textual references such as documents from Wikipedia (Thorne et al., 2018a, 2019); 2) the claim is verified against existing knowledge bases (Shi and Weninger, 2016; Syed et al., 2019). Both approaches assume their references are reliable. The first approach may limit evidence to only trusted resources such as Wikipedia, fact-checking websites, peer-reviewed academic papers, and government documents, achieving substantial coverage of information. However, the second approach faces bigger challenges in terms of coverage of

[2]The task of claim validation is referred to as fact-checking by some papers in the literature.

reliable information. Existing knowledge bases tend to be too small to cover sufficient information for claim validation purposes (Mendes et al., 2012; Azmy et al., 2018; Pellissier Tanon et al., 2020). Attempts have been made to automatically populate knowledge bases (Nakashole and Weikum, 2012; Adel, 2018; Balog, 2018; Mesquita et al., 2019) but this method has the risk of further introducing unreliable noise and makes it harder to maintain the knowledge bases. Due to its maturity and reliability, we focus on the first approach.

There have been a number of shared tasks focused on claim validation in slightly different ways. One of the major differences is whether the final verification step is reliant on previously identified pieces of evidence (such as Wikipedia documents or scientific articles) or it is instead reliant on the stances expressed by users (for example by aggregating supporting or opposing stances towards a story in social media). Of those relying on evidence, well-known shared tasks include FEVER (Thorne et al., 2018a) and SCIVER (Wadden et al., 2020), both of which perform evidence retrieval first and then perform claim verification based on that evidence. On the other hand, both UKP Snopes (Hanselowski et al., 2019) and RumourEval (Derczynski et al., 2017; Gorrell et al., 2018) proposed to tackle the task by retrieving texts relevant to a story, determining the stance of those texts afterwards, to ultimately classify the veracity value of the story.

2.2.2 Datasets

The NLP and AI community has developed valuable datasets to progress research in automated claim validation, though with common issues of being synthetic and imbalanced. As shown in Table 2.1, recent datasets are not only growing in size, but they also attempt to capture naturally occurring sentences, include context and metadata, offer evidence chains and cover different domains, languages and modality.

Name	# of Claims/Claim-Evidence Pairs	Domains	Details
PolitiFact (Vlachos and Riedel, 2014)	106 claims	Politics	Very small; metadata and evidence of various forms
Emergent (Ferreira and Vlachos, 2016)	300 claims	News	Very small; 2595 associated documents
LIAR (Wang, 2017)	12,836 claims	Politics	Medium; metadata

Snopes (Popat et al., 2017)	4,956 claims	Snopes website	Medium; 30 Google retrieved documents for each claim
FEVER (Thorne et al., 2018a)	185,445 claims	Wikipedia	Big; associated Wikipeida evidence
LIAR-PLUS (Alhindi et al., 2018)	12,836 claims	Politics	Medium; automatically extracted justifications
Perspectrum (Chen et al., 2019b)	907 claims	Debates	Small; evidence and perspectives
UKP Snopes (Hanselowski et al., 2019)	6,422 claims	Snopes website	Medium; associated evidence
MultiFC (Augenstein et al., 2019)	34,918 claims	Fact-checking websites	Medium; metadata and 10 Google retrieved webpages for each claim
SciFact (Wadden et al., 2020)	1,409 claims	Scientific papers	Small; associated documents
PolitiHop (Ostrowski et al., 2021)	500 claims	Politics	Very small; evidence chains for multi–hop reasoning
WikiFactCheck-English (Sathe et al., 2020)	124,821 claims	Wikipedia	Big; context and evidence
DanFEVER (Nørregaard and Derczynski, 2021)	6,407 claims	encyclopedia	Medium; Danish
ParsFEVER (Zarharan et al., 2021)	22,906 claims	Wikipedia	Medium; Farsi
X-Fact (Gupta and Srikumar, 2021)	31,189 claims	Fact-Checking websites	non-English claims from 25 languages with seven labels

Climate-FEVER (Diggelmann et al., 2021)	1,535 claims	Climate	Medium; 7,675 claim-evidence pairs with climate related claims verified against Wikipedia evidence
COVID-Fact (Saakyan et al., 2021)	4,086 claims	COVID-19	Medium; 1,296 supported claims from r/COVID19 subreddit and 2,790 automatically generated refuted claims
Vitamin-C (Schuster et al., 2021)	488,904 pairs	Wikipedia	Big; contrastive evidence from Wikipedia edits
FEVEROUS (Aly et al., 2021)	87,026 claims	Wikipedia	Biggest; evidence collected from both structured and unstructured information on whole Wikipedia
DialFact (Gupta et al., 2022a)	22,245 conversational claims	Wikipedia	Big; dialogue format
CHEF (Hu et al., 2022b)	10,000 claims	Fact-Checking websites	Medium; real-world claims collected from six Chinese fact-checking websites
MMM Gupta et al. (2022b)	10,473 claims	news	Medium; multilingual (Hindi, Bengali and Tamil); multimodal (text and image)
COVID-VTS (Liu et al., 2023)	10,000 claims	Twitter	Medium; multimodal (video, speech, claim); synthetic claims

Table 2.1: Claim Validation Datasets

2.2.3 Evidence Retrieval

Evidence retrieval is conventionally addressed in two steps: document retrieval and rationale selection. Document retrieval is the task of retrieving relevant documents that supports the prediction of a claim's veracity. Rationale selection is the task of selecting directly relevant sentences out of the retrieved documents to get final supporting evidence for claim verification.

Document Retrieval Deeply influenced by information retrieval research, the majority of work in the literature addresses it as a ranking problem consisting in retrieving the top k documents. Various combination of Named Entities, Noun Phrases and Capitalised Expressions from the claim were used to query search APIs such as Google or Wikipedia and search servers (Thorne et al., 2018b), when participating in the FEVER shared task. Metadata such as page viewership statistics is helpful to rank webpages (Nie et al., 2019). However, when search engines are not available, such as in the SCIVER shared task, the majority of effort goes into exploring similarity metrics that are used as a proxy to determine the documents' relevance to a claim. TF-IDF similarity is a common baseline (Wadden et al., 2020; Malon, 2018) and BM25 (Robertson et al., 1994) is demonstrated to be effective (Pradeep et al., 2021). When dealing with a specific domain, in-domain word embeddings are also a promising option, e.g. BioSentVec (Chen et al., 2019a) for the SCIFACT dataset (Li et al., 2021).

Instead of completely relying on unsupervised methods, improvements have been achieved by reranking based on supervised learning on top of a large number of retrieved documents (Pradeep et al., 2021).

Rationale Selection Keyword matching, sentence similarity scoring and supervised ranking are common approaches to rationale selection (Thorne et al., 2018b). Similar to document retrieval, attempts typically use one of these approaches or a combination of them to get a ranking score and select top k sentences as rationale with a manual choice of the k value (Pradeep et al., 2021).

Most of studies in the literature conduct evidence retrieval by addressing document retrieval and rationale selection in a pipeline manner, which ignores valuable information across sentences.

2.2.4 Claim Verification

Claim verification is commonly addressed as a text classification task by NLP researchers. Given a claim under investigation and its retrieved evidence, models need to reach a verdict of the claim, which may be 'SUPPORTS', 'REFUTES' or 'NOT_ENOUGH_INFO'. Some other datasets (Hanselowski et al., 2019; Wang, 2017) include other labels such as 'mostly-true', 'half-true', 'pants-fire', 'most false', 'most true' and 'other', whose finer granularity is more difficult to tackle through automated means and are sometimes collapsed into fewer labels. An important observation here is the difference in the types of labels used by different studies. Some studies rely on truth values (e.g. true, false, half-true), determining the veracity value of a claim. Others

refer to the concept of support instead (i.e. support, contradict), which instead determine whether there is an agreement between the claim and the reference. The latter avoids making an explicit connection with truthfulness, looking instead at the alignment of a claim with respect to a given reference.

The task of claim verification may be essentially addressed as a Recognising Textual Entailment (RTE) task, i.e. 'deciding, given two text fragments, whether the meaning of one text is entailed (can be inferred) from another text' (Dagan et al., 2009) or a Natural Language Inference (NLI) task, i.e. 'characterising and using semantic concepts of entailment and contradiction in computational systems' (Bowman et al., 2015).

Given that claim verification is predominantly addressed as a RTE or NLI task, we present a brief overview of them below. The RTE task, which dates back to 2005, focuses on detecting whether the hypothesis h is entailed by a given text t or not, which corresponds to 'SUPPORT' or not. Proposed models may take a linguistic approach, a statistical approach, a machine learning approach or a hybrid version of them. The NLI task, equipped with many large-scale labelled datasets, has powered large neural models to be the dominant approach. State-of-the-art models are large pre-trained language models that are fine-tuned on large NLI datasets.

Recognising Textual Entailment (RTE) Textual entailment is defined as a relation between a text T and a hypothesis H. Formal semantics defines that a text t entails hypothesis h if h is true in every possible circumstance where t is true (Chierchia and McConnell-Ginet, 2000). This definition, as well as many other formal linguistic theories, is theoretically sound but practically too rigid to handle uncertainty. In practical NLP context, we define entailment to include cases where the truth value of hypothesis h is highly plausible given text t, rather than absolutely certain (Dagan et al., 2009). In other words, 'text t entails hypothesis h if, typically, a human reading t would infer that h is most likely true' (Bar-Haim et al., 2014). In contrast to a formal theoretical definition, this somewhat informal definition of entailment allows and requires common sense background knowledge.

The task of RTE started as a two-way classification of deciding whether hypothesis h is entailed/supported by text t or not (Bar-Haim et al., 2006; Dagan et al., 2005; Giampiccolo et al., 2007). After the notion of 'contradiction', i.e., 'the negation of the hypothesis h is entailed by the text t' is introduced (de Marneffe et al., 2008), the RTE task became a three-way classification task of predicting labels of a text pair out of 'ENTAILMENT', 'CONTRADICTION' and

'UNKNOWN' (Giampiccolo et al., 2008).

Driven by a yearly RTE challenge from 2005 to 2011, the NLP community developed some useful datasets for the task, specifically the RTE1 - RTE7 datasets. Despite being relatively small and imbalanced, these datasets enabled the development and evaluation of various approaches. While lexical-based and syntax-based approaches struggled to achieve good results (Bar-Haim et al., 2014; Dagan et al., 2009), machine learning approaches achieved reasonable performance, often combined with logical or probabilistic methods.

One of the earlier models attempted to feed deep semantic features generated by first-order theorem prover and finite model builder into a machine learning classifier to make predictions (Bos and Markert, 2006). Surprisingly, the deep semantic features failed to outperform shallow semantic features. This is likely due to the models' naïve and rigid representation of sentences, lack of background knowledge and flawed sample distribution of the dataset.

Another intuitive approach is to first induce representations of text snippets into a hierarchical knowledge representation and then use a sound inferential mechanism to prove semantic entailment (De Salvo Braz et al., 2005). Despite its sound and intuitive system design, this model only achieved an overall accuracy of 65.9%.

Furthermore, the NatLog system deals with the problem in three stages. It first conducts linguistic analysis, then aligns the dependency graph of the text t and the hypothesis h, finally uses a decision tree classifier to perform entailment inference based on antonyms, polarity, graph structure and semantic relations (Chambers et al., 2007). This NatLog system trades low recall (31.71 on RTE3 test set) for higher precision (68.06 on RTE3 test set).

To help address the low recall achieved by first-order rules, the class of pair feature spaces was introduced (Zanzotto et al., 2009). It allowed the model to enrich the sentence-pair with 'placeholders' and then generate first-order rewrite rules to relax the rigidness. This model achieved around 68% overall accuracy on RTE3.

Moreover, COGEX developed a system that first transforms the text into three-layered semantically-rich logic form representations, then generates a set of linguistic and world knowledge axioms, and searches for a proof of entailment (Tatu and Moldovan, 2007). This system achieved an overall accuracy of 72.25%.

Overall, many inspiring hybrid models of logical inference and machine learning methods were developed for RTE challenges. Though they did not achieve perfect performance, we believe

they have great potential once equipped with better text representations and more powerful neural models.

Natural Language Inference (NLI) More recently, NLI is proposed as 'the problem of determining whether a natural language hypothesis *h* can reasonably be inferred from a given premise *p*' (Bowman et al., 2015; MacCartney and Manning, 2009). Noticeably, the definition of NLI is very similar to RTE and researchers tend to mention them together when addressing the problem.

Despite that, NLI datasets have improvements over RTE datasets. Earlier RTE datasets, published before the notion of 'CONTRADICTION' attracted enough attention, only have binary labelling of 'ENTAILMENT'. In contrast, NLI datasets all include three-way labelling that includes 'ENTAILMENT', 'CONTRADICTION' and 'UNKNOWN'. Furthermore, recent NLI datasets have larger size, more balanced label distribution and cover various domains. Table 2.1 presents NLI datasets that are potentially useful for claim verification.

With their large size and balanced design, NLI datasets have powered large neural network models, which has become the dominant approach. The common practice is to fine-tune a large pre-trained language model on the target NLI dataset, which may or may not be coupled with small task-specific techniques. Compared with traditional approaches, this approach improved text representations, achieved better generalisability and allowed more complex computing without relying on hand-crafted rules.

2.2.5 Discussion and Challenges

Despite the noticeable progress, current automated claim validation systems also face unique challenges and desire improvements over several key aspects: annotation reliance, system integrity, and model interpretability.

Annotation Reliance State-of-the-art systems heavily depend on fine-tuning pre-trained language models, necessitating extensive, high-quality labeled data that can be costly and impractical for certain niche or evolving domains. While recent datasets have made significant contributions, they often suffer from imbalance and synthetic nature, posing challenges for effective model training. While high-quality datasets are expected to further advance the field, there is a growing need for systems that can reduce reliance on vast amounts of labeled data and rapidly learn from a limited number of samples. Such systems would offer valuable flexibility and efficiency in addressing evolving information landscapes and domain-specific challenges.

System Scalability and Integrity Proposed automated claim validation systems cover a range of relevant tasks. Though a few of them try to jointly handle rationale selection and claim verification (Hidey and Diab, 2018; Li et al., 2021; Wadden et al., 2022), most of them are pipeline systems that train individual models to deal with subtasks separately. Improved scalability and integrity is desired. Fine-tuning pre-trained language models, the current dominant approach of various relevant tasks, requires lots of computing resources to train and inference. The scalability and accessibility of the proposed systems remain inferior.

Otherwise, increased system integrity is desired. Pipeline design has its inevitable disadvantage: downstream components can only make inferences on upstream results and errors accumulate throughout the pipeline. For instance, a claim verification component that takes the retrieved evidence and the detected claim as input will perform poorly with low-quality evidence or claims that are not checkable. Furthermore, the popular three-way classification approach may not be the best approach for claim verification. If a trained model struggles particularly to predict "REFUTES" class due to a lack of training data in this class, it may accumulate errors across classes. Moreover, current approaches leave limited space for aggregating evidence across sentences.

We believe a more compact overall system design is desired for automated claim validation such that it systematically handles subtasks. We believe a promising direction is to train a single model to handle all involved tasks for optimized overall performance.

Model Intepretability Neural networks are robust but struggle with interpretability and generalisability (Duan et al., 2020), which is of particular importance for automated claim validation. Underwhelming model interpretability may induce an increased probability of models making the right prediction based on the wrong reasoning. In contrast, symbolic systems that are unfortunately fragile and inflexible have strong interpretability and abstraction. Naturally, building a neural-symbolic system that integrates neural networks with symbolic logic becomes an interesting direction. In a nutshell, *neural-symbolic systems = connectionist machine + logical abstractions (Garcez et al., 2022).*

Researchers have proposed various architectures that incorporate first-order logic into neural networks. A recent study proposed a general framework capable of enhancing neural networks with declarative first-order logic (Hu et al., 2016). Another study explored a symbolic intermediate representation for neural surface realisation (Elder et al., 2019) that is similar to first-order logic.

Moreover, a recent attempt adapted module networks to model natural logic operations, which is enhanced with a memory component to model contextual information (Feng et al., 2020). Furthermore, RuleNN (Sen et al., 2020) is developed to tackle sentence classification where models are in the form of first-order logic, and achieved performance that is comparable to some neural models.

Neural-symbolic methods have a fascinating potential of attaining interpretability from symbolic models and robustness from neural models. Recently, ProoFVer (Krishna et al., 2022) introduces a seq2seq model to generate natural logic-based inferences as proofs for automated fact-checking. We believe such efforts on neural-symbolic methods for various tasks of automated fact-checking is promising and of particular interest to our society.

2.3 Few-Shot Claim Verification

As claim validation has emerged as a crucial task to combat the spread of false information, which can help fact-checking researchers and pratitioners as a fully automated tool that emphasise on classification performance (Thorne and Vlachos, 2018; Zeng et al., 2021), a classification tool with improved explainability (Kotonya and Toni, 2020; Guo et al., 2022), and a tool that focuses on assisting human workers (Nakov et al., 2021a). A body of natural language processing (NLP) research has investigated the task of claim verification: determining the veracity of a claim based on retrieved evidence. While the majority of previous work tackles the problem with fully supervised methods (Li et al., 2021; Zeng and Zubiaga, 2021; Zhang et al., 2021; Wadden et al., 2022; Rana et al., 2022b,a), deploying these methods face practicality issues. Emerging domains of misinformation often involve novel claims, limiting the availability of relevant labeled data. Fact-checkers often need to evaluate claims with time constraints, limiting the time allowed for conducting extensive fine-tuning of pretrained language models (PLMs). Hence, performing claim verification with limited labelled data is of particular importance in the real-world combat of misinformation.

In machine learning, few-shot learning is a framework where a model can effectively learn from a very small number of labeled samples, typically used when available training data is scarce. Specifically, with n-shot learning, we refer to the experimental setup where the training data has n samples per class. While the scarcity of annotations poses a major challenge to automated fact-checking (Zeng et al., 2021), research on few-shot learning techniques for claim verification

is limited to date.

To the best of our knowledge, previous literature that tackles few-shot claim verification only has one such attempt PB (perplexity-based method) (Lee et al., 2021). Lee et al. (2021) investigated a perplexity-based approach that solely relies on perplexity scores from PLMs. This method proved to achieve better performance on few-shot binary classification than fine-tuning a BERT model. However, their model was tested on binary claim verification, as opposed to the three-way NLI classification (Thorne and Vlachos, 2018) that claim verification is typically addressed as. Specifically, it has limited its applicability to binary claim verification, i.e., keeping the 'SUPPORTS' class and merging the 'REFUTES', and 'NOT_ENOUGH_INFO' classes into a new 'Not_SUPPORTS' class.

In contrast, our research introduced SEED (Zeng and Zubiaga, 2022), a method that calculates PLM-based pairwise semantic differences between claims and associated evidence. By deriving representative class vectors from these differences, SEED offers an efficient solution for few-shot claim verification and serves as one of our baseline models. While, the SOTA baseline for few-shot claim verification PB is limited to binary classification, SEED is also applicable to and experimented in three-class settings. The main application scenario of SEED is few-shot claim verification, but it may also apply to many other pairwise classification tasks such as natural language inference and stance detection. Due to its sensitive to data sampling within few-shot scenarios, SEED also offers the potential to be used for annotation quality evaluation as a good metric to determine whether the annotated data is of high quality or not with only a few samples. Similarly, SEED can be adapted to do task difficulty estimation: experiments show that SEED's few-shot performance on different datasets correlates well with how challenging the datasets are.

Another competitive training procedure for few-shot learning is PET (Schick and Schütze, 2021a,b). PET reformulates classification tasks into cloze tasks using templates. By calculating the probability of candidate tokens filling the placeholder [mask] position with an PLM, PET maps it to a preconfigured label. PET has shown competitive performance in a range of NLP classification tasks, but its adaptation to the context of automated fact-checking has not been studied. We therefore conduct few-shot claim verification experiments on it.

When addressing claim verification, both SEED and PET heavily rely on PLMs trained on NLI, which brings several limitations. Firstly, they face challenges when dealing with data that significantly differs from general NLI datasets, such as cases where the domain is highly

technical and different from general NLI data pairs and/or the evidence consists of large paragraphs rather than single sentences. Additionally, their reliance on NLI-trained models restricts their applicability to languages for which NLI datasets and corresponding PLMs are available, excluding their use in low-resource languages. To address these limitations, our research further proposed MAPLE (Zeng and Zubiaga, 2024), which does not rely on NLI-trained models but instead utilizes unlabelled claim-evidence pairs which could be abundant and useful for domain adaptation.

In addition, recent advancements in generative LLMs with multi-billion parameters have showcased impressive few-shot capabilities. However, closed-source pioneering models, including GPT-3.5 and GPT-4, present reproducibility challenges with their behavior changing over time (Chen et al., 2024). In this study, we prioritize open-source solutions, with a particular focus on LLaMA 2, a recent model that surpasses existing open-source alternatives across various benchmarks (Touvron et al., 2023). The primary drawback of these approaches lies in their requirement for advanced computational infrastructure, a substantial computational budget, and extended inference times. MAPLE tackles these constraints by utilizing parameter-efficient models, aiming to improve both resource and runtime efficiency. Experiments show that MAPLE is robust to noisy and challenging data in realistic fact-checking scenarios, including scenarios when oracle evidences are absent. With an efficient integration workflow, the application of MAPLE in real-world scenarios can bring in a decent claim verification tool to assist fact-checkers in combating emerging domains of misinformation, with minimal cost in annotation and computational resources.

While few-shot claim verification experiments are often conducted with random sampling on the supervisory data, it is not a hard requirement in real-work fact-checking. As availability of detected claims can be abundant, creating lots of claim-evidence pairs with automated retrieval methods, i.e., unlabelled data for the claim verification task, is very practical. To maximise the effectiveness of the human annotations, we aim to study data annotation priotisation in order to select the most beneficial data, with active learning strategies.

To the best of our knowledge, however, no work has investigated the use of active learning in the context of claim verification. To further research in this direction, our research proposes Active PETs (Zeng and Zubiaga, 2023), a model that incorporates active learning capabilities into PET. Experiments demonstrate Active PETs' significant improvements over random sampling and other active learning strategies, particularly when the data distribution is highly skewed.

Fact-checking researchers and practitioners may incorporate it into their data annotation pipeline and use it to select optimal samples from the unlabelled pool for optimal usage of annotation budgets, especially when the unlabelled data pool is expected to have highly imbalanced label distribution.

2.4 Methodological Foundations

In this section, we explore the methodological foundations of the development of our proposed methods: SEED, MAPLE, and Active PETs. Each subsection below delves into the literature relevant to our novel approaches, shedding light on the progression of ideas and technologies that have shaped our contributions to the field. This includes examining the utilization of representative vectors for text classification, which underlies our SEED method; dissecting the role of natural language generation (NLG) metrics and the concept of language evolution in informing our MAPLE approach; and scrutinizing the evolving landscape of Active Learning strategies that enrich our understanding and implementation Active PETs. Through this exploration, we aim to highlight the relevance and novelty of our contributions against the backdrop of existing research.

2.4.1 Representative Vectors for Text Classification

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 in Chapter 4 further proposes the use of semantic differences to simulate a meaningful and comparable representation of claim-evidence pairs, enabling its application on the task of claim verification.

2.4.2 NLG Metrics and Understanding Language Evolution

NLG metrics NLG evaluation metrics play a crucial role in evaluating the quality of generated texts. Classic metrics such as BLEU (Bilingual Evaluation Understudy) (Papineni et al., 2002), ROUGE (Recall-Oriented Understudy for Gisting Evaluation) (Lin, 2004), and METEOR (Metric

for Evaluation of Translation with Explicit ORdering) (Banerjee and Lavie, 2005) remain as the most widely used metrics. They address the evaluation as a matching task, quantifying n-gram overlap with recall, precision and F-score and providing lexical-level evaluations.

Recent advancements include SacreBLEU (Post, 2018), which enhances reproducibility, tokenization support, and ease of statistical significance reporting. In contrast, BLEURT (Bilingual Evaluation Understudy with Representations from Transformers) (Sellam et al., 2020) advances semantic-level evaluations and treats evaluation as a regression task using PLMs. Another metric, BARTScore (Yuan et al., 2021), approaches evaluation as a text generation task for LLMs, calculating the BARTScore as the weighted log probability of one text given another text.

Given our primary interest in the semantic shift during pairwise language evolution, our research in Chapter 5 proposes 'SemSim' as an alternative metric to evaluate NLG performance.

Understanding Language Evolution Language evolution has been the subject of several theories, including biological evolution, learning, and cultural evolution (Lekvam et al., 2014). Studies conducted in laboratory settings have explored the intricate nature of various phenomena, offering valuable insights into the emergence of language (Scott-Phillips and Kirby, 2010).

Researchers have focused on modeling evolution within language families to identify patterns in phonetic features across observed languages Nouri and Yangarber (2016). Computational research has also introduced tools such as language evolution simulators, examining word-level evolution within language families (Ciobanu and Dinu, 2018), and realistic geographic environments to simulate language and linguistic feature development over time (Kapur and Rogers, 2020). These studies tackle various related issues for historical linguistics, areal linguistics, and linguistic typology.

While language evolution research often adopts a macro and historical perspective, our research engages in micro-level analysis, i.e. asking "what path does it take for a piece of text to migrate into another piece". Interestingly, the convergence process during seq2seq training simulates such a path of evolving or transitioning language. In our work in Chapter 5, we investigate language transition across seq2seq training epochs and further utilize it to conduct pairwise classification.

2.4.3 Active Learning

Active Learning (AL) is a paradigm used where labelled data is scarce (Ein-Dor et al., 2020). The key idea is that a strategic selection of training instances to be labelled can lead to improved performance with less training (Settles, 2009). Active learning methods are provided with an unlabelled pool of data, on which a querying step is used to select candidate instances to be annotated with the aim of optimising performance of a model trained on that data. The goal is therefore to optimise performance with as little annotation –and consequently budget– as possible. Traditional active learning query strategies mainly include uncertainty sampling, query-by-committee (QBC) strategy, error/variance reduction strategy and density weighted methods (Settles, 2012). Recent empirical studies have revisited the traditional strategies in the context of PLMs: Ein-Dor et al. (2020) examined various active learning strategies with BERT (Devlin et al., 2019), though limited to binary classification tasks. Schröder et al. (2022) conducted experiments with ELECTRA (Clark et al., 2020), BERT, and DistilRoBERTa (Sanh et al., 2019) respectively, while limiting the scope to uncertainty-based sampling.

Recent efforts on combining active learning with PLMs go into both warm-start and cold-start strategies. Warm-start strategies require a small initial set of labelled data to select additional instances, while cold-start strategies can be used without an initial set of labelled data. Ash et al. (2020) proposed Batch Active learning by Diverse Gradient Embeddings (BADGE) that samples a batch of instances based on diversity in gradient loss. Margatina et al. (2021) proposed Contrastive Active Learning (CAL), the state-of-the-art (SOTA) warm-start strategy that highlights data with similar feature space but maximally different predictions. Furthermore, Active Learning by Processing Surprisal (ALPS) (Yuan et al., 2020), the SOTA cold-start strategy, utilises masked language model (MLM) loss as an indicator of model uncertainty. Our research uses BADGE, CAL and ALPS for baseline comparison, please see detailed descriptions in section 6.2.

To the best of our knowledge, QBC strategies (Seung et al., 1992; Dagan and Engelson, 1995; Freund and Haussler, 1997) that utilise a committee of models remains to be explored with PLMs, as previous studies limit their scope at measuring single model uncertainty. Nowadays various PLMs are publicly available that applying an ensemble-based query strategy on a downstream task becomes realistic, especially in few-shot settings where the computation required is relatively cheap. Furthermore, previous studies always perform fine-tuning to get classification results from PLMs. Our work in Chapter 6 presents the first attempt at integrating an active learning strategy

into PET, a few-shot learning claim verification method that is suitable for a wide range settings on the number of labelled data.

2.5 Summary

In this chapter, we have traversed the evolving landscape of automated fact-checking, marking its significance against the backdrop of digital misinformation. From foundational challenges to innovative solutions in claim validation and few-shot claim verification, our discussion has not only highlighted the field's complexity but also its critical role in leveraging limited data for reliable inference. By anchoring our research in the methodological advances of natural language processing and artificial intelligence, we pave the way for our thesis's novel contributions.

Chapter 3

Shared Experimental Resources

This chapter delineates the comprehensive experimental framework underpinning the research presented in this thesis. It provides a detailed examination of the datasets, baselines, problem formulation, and evaluation metrics used in the experiments conducted with each of the proposed novel methods: SEED, MAPLE, and Active PETs.

3.1 Datasets

Here, we delve deeper into the characteristics and significance of the three pivotal datasets utilized in our experiments: FEVER, Climate FEVER, and SciFact. Each dataset has been carefully selected for its unique attributes and contributions to the field of fact-checking, ranging from the groundbreaking scope of FEVER as the initial large-scale dataset, to the specialized focus and challenging nature of Climate FEVER and SciFact. To offer a comprehensive understanding of the experimental framework, we will provide an overview of each dataset, highlighting its relevance. Additionally, we will illustrate the specifics of our experimental datasets through showcasing representative data samples from each dataset, and detailing the label distributions for each datasets.

3.1.1 Dataset Profiles

FEVER FEVER (Thorne et al., 2018a) is a large-scale dataset for automated fact-checking. It contains claims that are manually modified from Wikipedia sentences along with their corresponding Wikipedia evidences. Despite criticisms of its synthetic nature by researchers in

the fact-checking domain, it has been widely used also for other tasks outside of fact-checking. Various NLP benchmarks, such as KILT (Petroni et al., 2021), include the claim verification task of FEVER to test models' reasoning capabilities. To enable direct comparison with the baseline model PB (Lee et al., 2021), FEVER is used in claim verification experiments in chapter 4 and 5, following the practice of using oracle evidence (Lee et al., 2021; Petroni et al., 2021) . As claims from FEVER dataset are synthetic mutations from evidence texts, they are synthetic and lexically very close to their evidence. Hence, FEVER does not provide realistic scenarios and is not used in active learning experiments in chatper 6, which focuses on solving realistic challenges for data annotation priotisation. We only use the test set of the original FEVER dataset, as it contains higher-quality data and the quantity is sufficient for few-shot experiments. We reserve 150 instances for each class to form a test set and leave the rest in the train set.

cFEVER Climate FEVER (Diggelmann et al., 2021) is a challenging, large-scale dataset that consists of claim and evidence pairs related to climate change, along with their veracity labels. Since the dataset does not naturally provide options for setting up retrieval modules, we directly use it for the claim verification task. Similarly, we reserve 150 instances for each class to form a test set and leave the rest in the train set.

SciFact SciFact (Wadden et al., 2020) provides scientific claims with their veracity labels, along with a collection of scientific paper abstracts, some of which contain rationales to resolve the claims. Additionally, it provides oracle rationales that can be linked to each claim. Research on SciFact places strong emphasis on the evidence retrieval module. Hence, we conduct experiments on SciFact with two configurations: SciFact_oracle and SciFact_retrieved. The former utilizes oracle evidence provided by the annotations, while the latter uses evidence retrieved by a retrieval model, namely BM25, to retrieve the top 3 abstracts as evidences (Wadden et al., 2022; Zeng and Zubiaga, 2023). We merge the original SciFact train set and dev set and redistribute the data to form a test set that contains 150 instances for each class, using the rest as the train set.

3.1.2 Dataset Samples

In Table 3.1, we showcase representative samples from each dataset to illustrate the diversity and nature of the data our experiments engage with. These samples provide insights into the challenges and considerations unique to each dataset, highlighting the varied contexts in which our methods are applied.

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."	*'SUPPORTS'*
"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."	*'REFUTES'*
"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."	*'NOT_ ENOUGH_ INFO'*

cFEVER		
Claim	**Evidence**	**Veracity**
"Coral atolls grow as sea levels rise."	"Gradual sea-level rise also allows for coral polyp activity to raise the atolls with the sea level."	*'SUPPORTS'*
"There's no trend in hurricane-related flooding in the U.S."	"Widespread heavy rainfall contributed to significant inland flooding from Louisiana into Arkansas."	*'REFUTES'*
"The warming is not nearly as great as the climate change computer models have predicted."	"The model predicted <0.2 °C warming for upper air at 700 mb and 500 mb."	*'NOT_ ENOUGH_ INFO'*

SCIFACT_oracle

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."	*'SUPPORTS'*
"Gene expression does not vary appreciably across genetically identical cells."	"Genetically identical cells sharing an environment can display markedly different phenotypes."	*'REFUTES'*
"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."	*'NOT_ ENOUGH_ INFO'*

SCIFACT_retrieved

Claim	Evidence	Veracity

"Neutrophil extracellular trap (NET) antigens may contain the targeted autoantigens PR3 and MPO."	"Netting neutrophils in autoimmune small-vessel vasculitis Small-vessel vasculitis (SVV) is a chronic autoinflammatory condition linked to antineutrophil cytoplasm autoantibodies (ANCAs). Here we show that chromatin fibers, so-called neutrophil extracellular traps (NETs), are released by ANCA-stimulated neutrophils and contain the targeted autoantigens proteinase-3 (PR3) and myeloperoxidase (MPO). Deposition of NETs in inflamed kidneys and circulating MPO-DNA complexes suggest that NET formation triggers vasculitis and promotes the autoimmune response against neutrophil components in individuals with SVV."	*'SUPPORTS'*
"Cytochrome c is transferred from cytosol to the mitochondrial intermembrane space during apoptosis."	"At the gates of death. Apoptosis that proceeds via the mitochondrial pathway involves mitochondrial outer membrane permeabilization (MOMP), responsible for the release of cytochrome c and other proteins of the mitochondrial intermembrane space. This essential step is controlled and mediated by proteins of the Bcl-2 family. The proapoptotic proteins Bax and Bak are required for MOMP, while the antiapoptotic Bcl-2 proteins, including Bcl-2, Bcl-xL, Mcl-1, and others, prevent MOMP. Different proapoptotic BH3-only proteins act to interfere with the function of the antiapoptotic Bcl-2 members andor activate Bax and Bak. Here, we discuss an emerging view, proposed by Certo et al. in this issue of Cancer Cell, on how these interactions result in MOMP and apoptosis."	*'REFUTES'*

"Incidence of heart failure increased by 10% in women since 1979."	"Clinical epidemiology of heart failure. The aim of this paper is to review the clinical epidemiology of heart failure. The last paper comprehensively addressing the epidemiology of heart failure in Heart appeared in 2000. Despite an increase in manuscripts describing epidemiological aspects of heart failure since the 1990s, additional information is still needed, as indicated by various editorials."	*'NOT_ ENOUGH_ INFO'*

Table 3.1: Data samples for each dataset.

3.1.3 Dataset Label Distribution

Table 3.2 details the label distributions for the datasets, offering a quantitative glimpse into the class balance—or lack thereof—within each. This overview is pivotal for understanding the datasets' inherent challenges and complexities.

Table 3.2: Unlabelled pool label distribution for each dataset.

	FEVER	**cFEVER**	**SciFact_oracle**	**SciFact_retrieved**
'SUPPORTS'	3099	1789	356	266
'REFUTES'	3069	652	115	61
'NOT_ENOUGH_INFO'	3183	4778	294	2530
Total unlabelled pairs	9351	7219	765	2857

3.2 Baselines

This section outlines the baseline methods against which the proposed novel methods are benchmarked. It includes a perplexity-based approach, Pattern Exploiting Training (PET), and the utilization of Large Language Model Meta AI 2 (LLaMA 2), each offering a distinct perspective on claim verification and few-shot learning performance. [1]

[1] In this section, we only list the claim verification baselines used for SEED and MAPLE experiments. For active learning baselines, please see in section 6.2.

Perplexity-based Approach

(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 'SUPPORTS' and 'Not_SUPPORTS': if the score is lower than the threshold th, it is classified as 'Not_SUPPORTS' and otherwise 'SUPPORTS'.

Pattern Exploiting Training

Pattern Exploiting Training (PET) (Schick and Schütze, 2021a,b) is a semi-supervised training procedure that can reformulate various classification tasks into cloze questions with natural language patterns and has demonstrated competitive performance in various few-shot classification tasks. To predict the label for a given instance x, it is first reformulated into manually designed patterns that have the placeholder $[mask]$. Then, the probability of each candidate token for replacing $[mask]$ is calculated by using a pretrained language model, where each candidate is mapped to a label according to a manually designed verbaliser.

Large Language Model Meta AI 2

Large Language Model Meta AI (LLaMA) 2 (Touvron et al., 2023) is a recent generative LLMs with multi-billion parameters that uses an optimized transformer architecture. After pertaining, it further went through supervised fine-tuning (SFT) and reinforcement learning with human feedback (RLHF) for improved helpfulness and safety. As an open-source alternative to ChatGPT, it has impressive few-shot learning capabilities via simple prompting.

3.3 Problem Formulation

Our research investigates claim verification tasks, focusing on samples comprised of a claim and its supporting evidence, alongside the veracity label annotated to this pair. The experimental setup includes a training set with labeled data for model learning, a test set for result evaluation with a balanced label distribution, and, optionally, an unlabeled data pool distinct from both training and test sets.

Few-Shot Learning Framework The experiments pivot around the "K-way, N-shot" paradigm, tailored for few-shot claim verification, with k representing the classification problem's complexity (binary or three-way) and n denoting the number of labeled examples per class in the training set. For SEED, we explore a range of n in the settings of 2, 4, 6, 8, 10, 20, 30, 40, 50, to 100 shots,

facilitating a comprehensive comparison against previous state-of-the-art (SOTA) methods in both binary and three-way classifications. MAPLE's investigation predominantly addresses three-way settings, advancing beyond the perplexity-based baseline significantly surpassed by SEED. SEED and MAPLE experiments are iterated 10 and 100 times, respectively, to ensure reliability.

Adaptation for Active Learning Active PETs introduces a novel approach within the few-shot learning domain, focusing on a three-way classification framework ($k = 3$) and adapting to the nuances of active learning. Unlike traditional settings, the training set evolves through an active learning sampling strategy, starting from a zero-shot scenario. This process iteratively selects i new samples for annotation, continuously refining the model with each batch, until reaching a maximum of m instances. In our experiments, $i = 10$, $m = 300$. This adaptation acknowledges the variable nature of labeled data in active learning contexts, diverging from the per-class instance count typical in few-shot learning.

3.4 Evaluation Metrics

We employ accuracy and macro F1 as the main metrics to evaluate the effectiveness of few-shot claim verification techniques. Given the inherent variability in repeated experiments, our focus is on reporting mean accuracy for SEED and mean macro F1 score for MAPLE to guarantee the reliability of our findings.

To provide a deeper insight into the variability and reliability of these results, standard deviations for both metrics are presented in the Appendix C.1 for all methods in 5-shot settings. This approach allows us to assess the consistency and stability of our models across various experimental conditions, offering a more comprehensive understanding of their performance in few-shot learning contexts.

Additionally, we analyze classwise F1 scores for MAPLE within 5-shot scenarios in Appendix C.2, offering detailed insights into the model's discriminative power across different claim categories: 'SUPPORTS', 'REFUTES', and 'NOT_ENOUGH_INFO'. This nuanced analysis helps identify specific strengths and areas for enhancement.

Active PETs, employing a unique active learning strategy, diverges from the experimental framework of repeated experiments used in SEED and MAPLE. As there is no repeated experiments due to its distinctive approach, we directly report the macro F1 performance for Active PETs.

3.5 Summary

In this chapter, we have presented detailed information on the shared experimental resources, paving the way for the following chapters. Our study delves into few-shot claim verification methods primarily within the “K-way, N-shot" framework, with adjustments made for active learning experiments. We scrutinize SEED, MAPLE, and Active PETs against tailored baselines, assessing their effectiveness in varied settings—SEED in binary and three-way, and MAPLE and Active PETs in three-way scenarios. Evaluations utilize accuracy and macro F1 scores to gauge classifier performance, adhering to machine learning evaluation standards. Through meticulous experimental design, our research endeavors to demonstrate the promise of few-shot learning in overcoming data limitations for automated fact-checking.

Chapter 4

SEED: Aggregating Pairwise Semantic Differences for Few-Shot Claim Verification

---

In this chapter, we hypothesize that a method can leverage a small number of training instances, such that the semantic differences will be similar within 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 4.1 provides an illustration: 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.

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. 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 verification 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 claim verification task based on semantic differences, which we consistently demonstrate in three different settings on two datasets.

We make the following contributions:

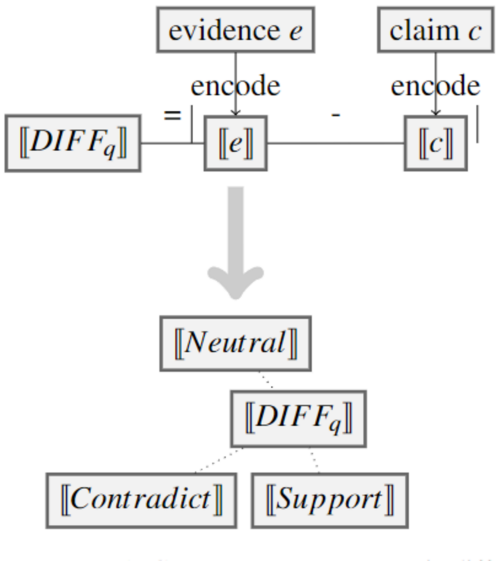

Figure 4.1: SEED Illustration.

- We introduce SEED, a novel method that computes semantic differences within claim-evidence pairs for effective and efficient few-shot claim verification.
- By experimenting on two datasets, we demonstrate the effectiveness of SEED to outperform two competitive baselines in the most challenging settings with a limited number of shots. While the state-of-the-art perplexity-based model is restricted to binary classification, SEED offers the flexibility to be used in two- or three-class settings. By looking at classwise performance results, we further demonstrate the consistent improvement of SEED across all classes.
- 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.

4.1 Methodology

We hypothesise that we can make use of sentence embeddings 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

pair made of *claim* and *evidence*, we first leverage a pre-trained language model through sentence-transformers library (Reimers and Gurevych, 2019) to obtain sentence embeddings $[\![claim]\!]$ and $[\![evidence]\!]$. Specifically, embeddings are obtained by conducting mean pooling with attention mask over the last hidden state. We then capture a representation of their semantic difference by calculating the element-wise difference $|[\![claim]\!] - [\![evidence]\!]|$. To the best of our knowledge, its previous implementation is only found in (Reimers and Gurevych, 2019) as one of many available classification objective functions, leaving room for further exploration. Formally, for a claim-evidence pair i that has $evidence_i$ and $claim_i$, we have equation 4.1:

$$[\![DIFF_i]\!] = |[\![evidence_i]\!] - [\![claim_i]\!]| \tag{4.1}$$

To address the task of claim verification 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 4.2.

$$\begin{aligned} &[\![Relation_c]\!] \\ &= \overline{[\![DIFF_c]\!]} \\ &= \frac{1}{n}\sum_{i=1}^{n}([\![DIFF_i]\!]) \\ &= \frac{1}{n}\sum_{i=1}^{n}(|[\![evidence_i]\!] - [\![claim_i]\!]|) \end{aligned} \tag{4.2}$$

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. $[\![SUPPORTS]\!]$, $[\![REFUTES]\!]$ and $[\![NOT\_ENOUGH\_INFO]\!]$ for three-way claim verification, and finally inherit the veracity label from the candidate relation vector that has the smallest Euclidean distance value.

4.2 Experimental Settings

Here we focus the experiments on the FEVER (Thorne et al., 2018a) and SciFact_oracle (Wadden et al., 2020) datasets configurations (see examples in Section 3.1.2), as they are the more influential datasets. For comprehensive experimental results on all presented dataset configuration, see

appendix C.1. Apart from three-way classification on both configurations, we also conduct experiments on binary FEVER to enable direct comparison with previous SOTA PB.

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) uses conditional perplexity scores generated by pre-trained language models to find a threshold that enables binary predictions. 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 [1]. 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.

SEED We implement SEED using the sentence-transformers library (Reimers and Gurevych, 2019) and the 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. [2] We include experiments with $SEED_{BERT_{NLI}}$ due to the proximity between the claim verification and natural language inference tasks. We use $SEED_{BERT_B}$, $SEED_{BERT_L}$ and $SEED_{BERT_{NLI}}$ to denote them hereafter.

Experimental Setup Experiments are conducted in three different configurations: binary FEVER claim verification, three-way FEVER claim verification and three-way SciFact_oracle claim verification. 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. 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 'SUPPORTS' instances and 3333 'Not_SUPPORTS' instances.[3] For n-shot settings, we sample n instances per class as the train set, and use $3333 - n$

[1]The associated model ids from huggingface model hub (Wolf et al., 2020) are *bert-base-uncased*, *bert-large-uncased*, *roberta-base* and *roberta-large* respectively

[2]The first two are available on huggingface model hub (Wolf et al., 2020) 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-transformers repository with model id *bert-base-nli-mean-tokens*.

[3]The 'Not_SUPPORTS' is obtained by sampling and merging original instances from both 'REFUTES' and 'NOT_ENOUGH_INFO' by the original authors of the PB method (Lee et al., 2021). We inherit the

instances per class as the test set. We present experiments with all three methods (SEED, PB, FT).

We conduct n-shot experiments (*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 trends 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 the performance fluctuations owing to the randomness of shots selection, we report the mean results for each n-shot experiment obtained by using 10 different random seeds ranging from 123 to 132. 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.

4.3 Results

We first report overall accuracy performance of each task formulation, then report classwise F1 scores for three-way task formulations. Finally we report statistical significance results.

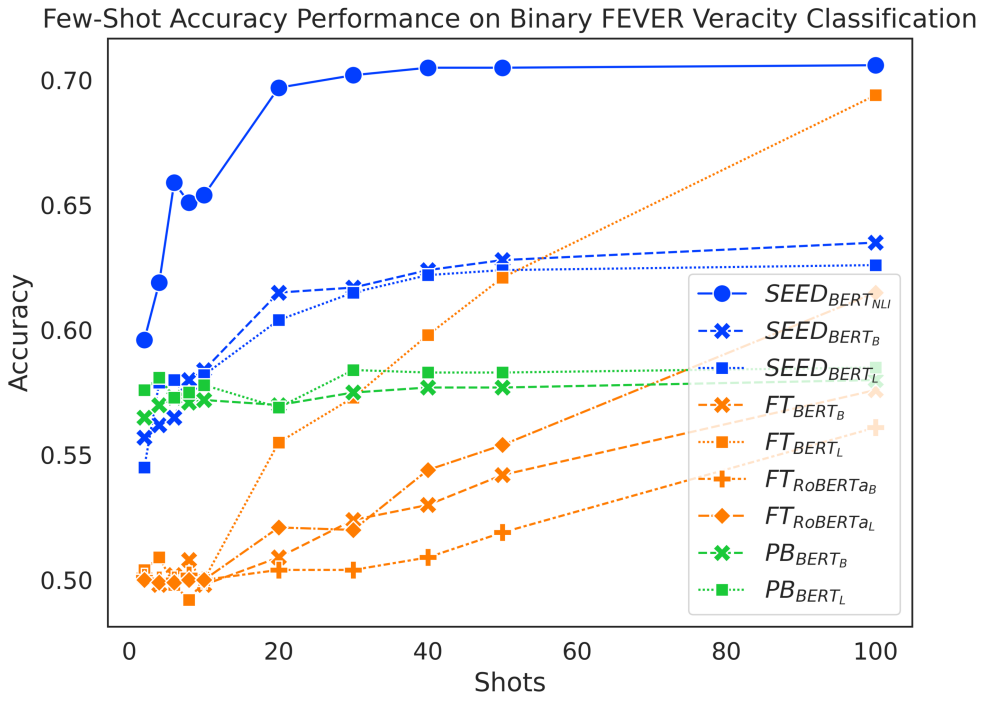

Figure 4.2: Comparison of few-shot accuracy performance on the binary FEVER dataset.

data files from previous research.

FEVER Binary Classification As shown in Figure 4.2, SEED achieves the overall best performance in few-shot settings. It suggests positive answers to our research questions: sentence embeddings from pretrained language models can be effectively utilised to compute semantic differences between claim-evidence pairs and they do contribute positively to the task of claim verification 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, which in turn outperforms FT. In 40-shot and 50-shot settings, 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 from 10-shot to 50-shot settings. In addition, $SEED_{BERT_{NLI}}$ always achieves the best performance up to 100 shots.

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 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.

FEVER Three-Way Classification Figure 4.3 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 reaches its performance peak at around 60% and $SEED_{BERT_{NLI}}$ peaks at around 68%. Given 30 or more

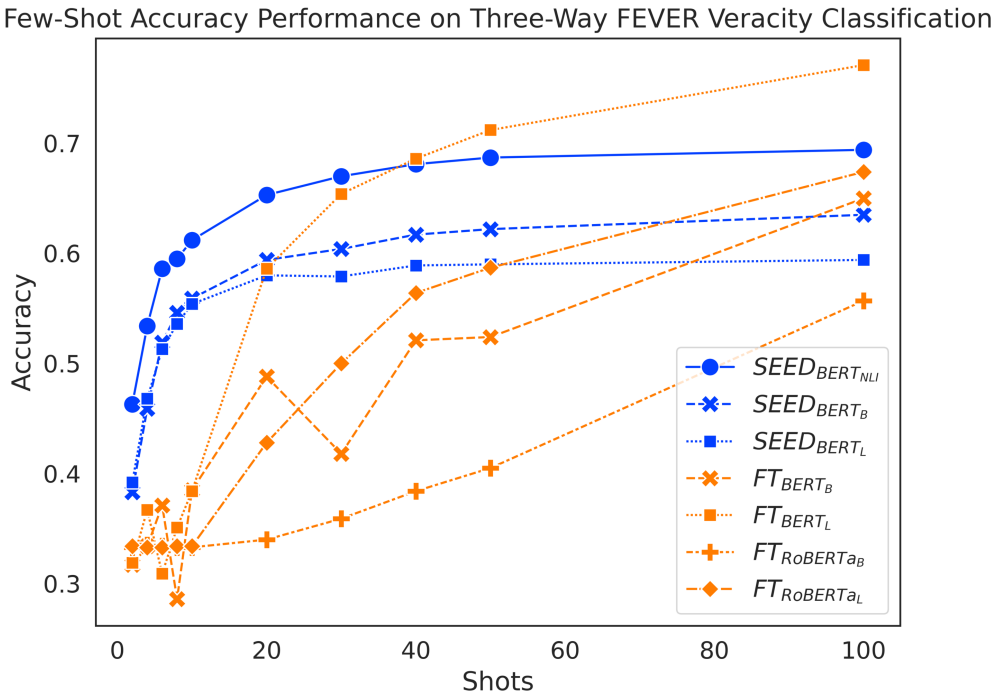

Figure 4.3: Comparison of few-shot accuracy performance on the FEVER dataset.

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 substantial 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. Like performance on binary FEVER, performance on three-way FEVER also suggests positive answers to our research questions: semantic differences between claim-evidence pairs can be captured by utilising sentence embeddings and positive contributions to the task of claim verification in few-shot settings are observed.

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.

SciFact_oracle Three-Way Classification Figure 4.4 shows again an expected increase in performance for both methods as they use more training data. Despite taking a bit longer to pick up,

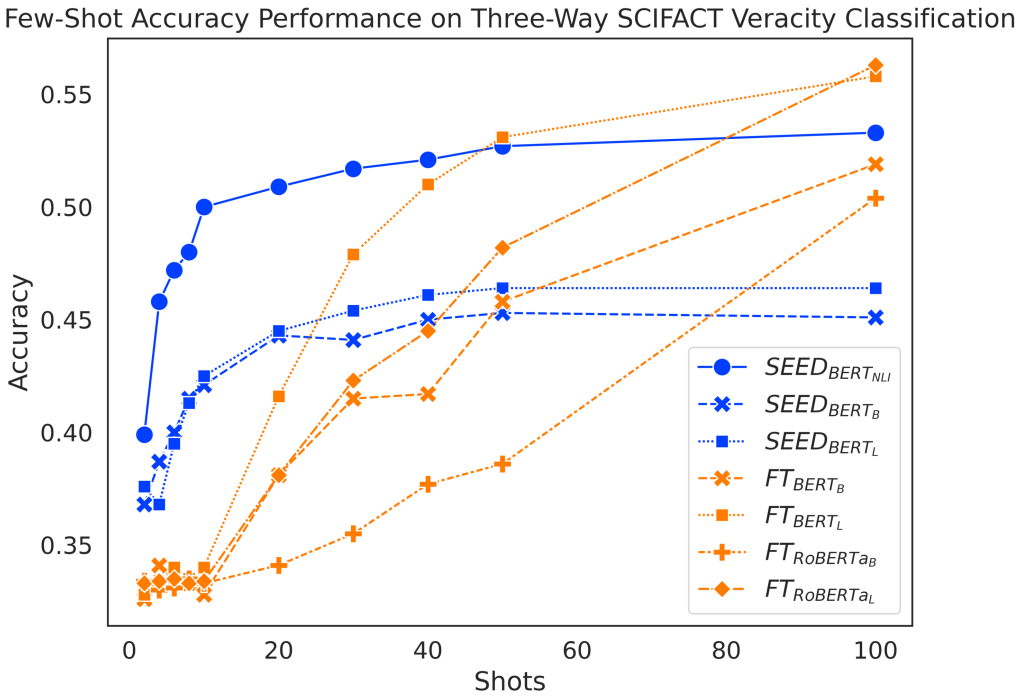

Figure 4.4: Comparison of few-shot accuracy performance on the SciFact_oracle dataset configuration.

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 claim verification. $SEED_{BERT_B}$ 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. Similarly, performance on SciFact_oracle dataset configuration leads to positive answers to our research questions: sentence embeddings from pretrained language models can be effectively utilised to compute semantic differences and make positive contributions to few-shot claim verification task.

The accuracy scores on the SciFact_oracle dataset configuration 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_oracle than on FEVER.

This highlights the challenging nature of the SciFact dataset, where SEED still remains the best in few-shot settings.

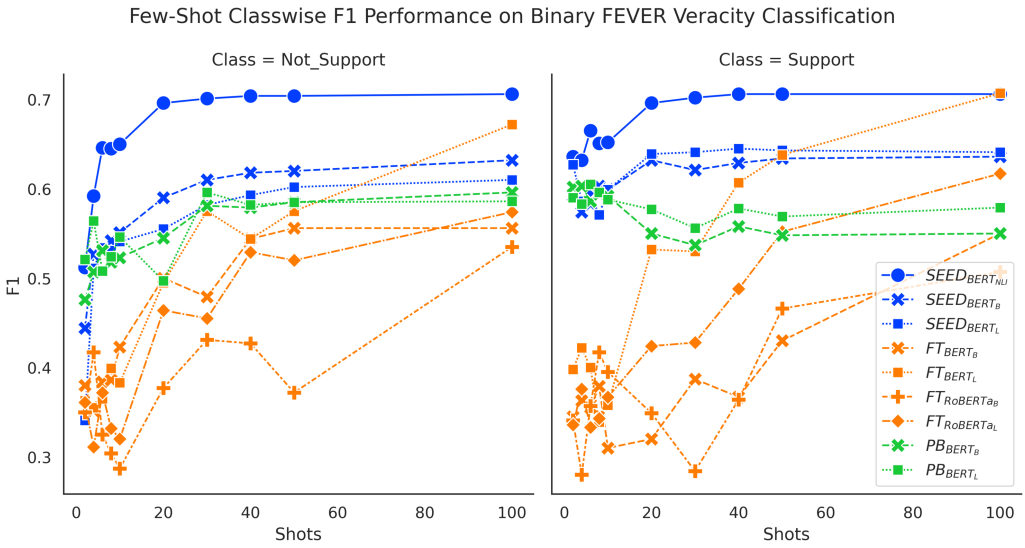

Figure 4.5: Comparison of few-shot classwise F1 performance on the binary FEVER dataset.

Classwise F1 Performances We present classwise F1 performance here for further understanding of the results. Figure 4.5 sheds light on addressing the task of FEVER binary claim verification. 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 'SUPPORTS' class, when given 10 or fewer shots. Despite that the PB method initially achieves very high performance on the 'SUPPORTS' class at around 60%, it then experiences a performance drop and ends at around 55% for BERT-base and 58% for BERT-large.

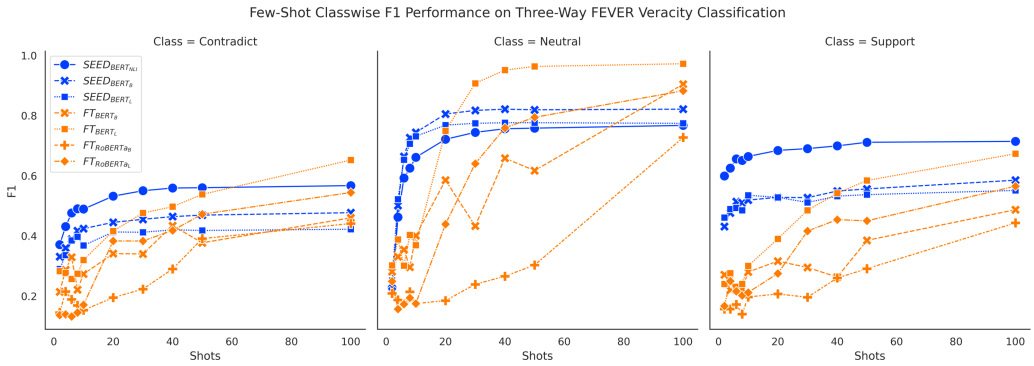

Figure 4.6: Comparison of few-shot classwise F1 performance on the FEVER dataset.

Figures 4.6 and 4.7 show consistent classwise performance patterns in tackling three-way

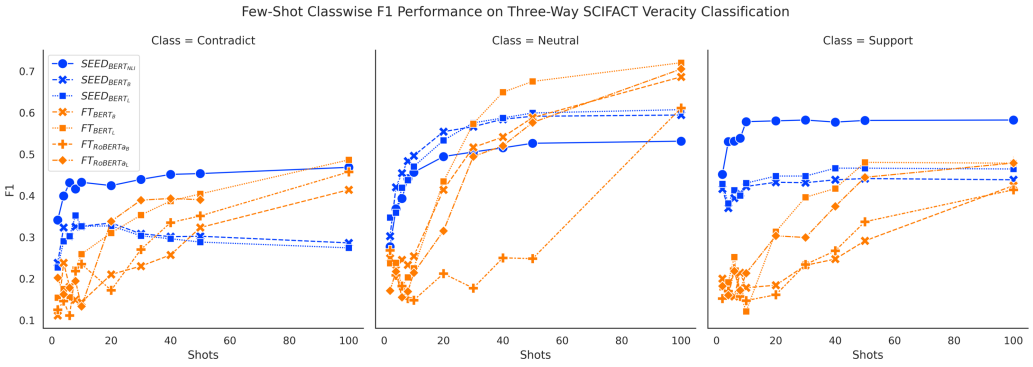

Figure 4.7: Comparison of few-shot classwise F1 performance on the SciFact_oracle dataset configuration.

claim verification on both FEVER and SciFact_oracle. Both figures indicate that SEED has better overall performance in all three classes when given fewer than 20 shots, where performance on the 'SUPPORTS' class always has absolute advantages over the FT method and performance on the 'NOT_ENOUGH_INFO' class experiences the biggest boost. At around 20-shots the FT method starts to overtake largely due to improved performance on the 'NOT_ENOUGH_INFO' class. Interestingly, within SEED, $SEED_{BERT_B}$ outperforms $SEED_{BERT_L}$, which in turn outperforms $SEED_{BERT_{NLI}}$.

Furthermore, classwise F1 performance also sheds light on the interesting SEED performance difference noted previously: $SEED_{BERT_B}$ outperforms $SEED_{BERT_L}$ in three-way claim verification with noticeable margin on FEVER three-way claim verification. Figure 4.6 shows that $SEED_{BERT_B}$ has clear performance advantages over $SEED_{BERT_L}$ on the 'REFUTES' and 'NOT_ENOUGH_INFO' classes on FEVER three-way claim verification, which may be the main cause of the performance difference. When conducting binary claim verification on FEVER where the 'REFUTES' and 'NOT_ENOUGH_INFO' classes are merged together, the performance advantages from $SEED_{BERT_B}$ over $SEED_{BERT_L}$ are trivial. Otherwise, $SEED_{BERT_B}$ does not outperform $SEED_{BERT_L}$ on the SciFact_oracle dataset configuration as shown in Figure 4.4. Meanwhile, Figure 4.7 does not demonstrate $SEED_{BERT_B}$'s performance advantages on distinguishing the 'REFUTES' and 'NOT_ENOUGH_INFO' classes on SciFact_oracle. We conjecture that $SEED_{BERT_B}$ is better at capturing simple differences between 'REFUTES' and 'NOT_ENOUGH_INFO' classes while $SEED_{BERT_L}$ is better at capturing complex differences due to their size difference. Given that FEVER is a synthetically generated dataset, it is to be expected that it includes more cases of simpler differences.

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 'SUPPORTS' 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 'NOT_ENOUGH_INFO' class.

Statistical Significance We present statistical significance test results conducted based on McNemar's Test to demonstrate robustness of SEED, compared with FT. For demonstration purposes, results are calculated in 20-shot setting with the sampling seed set as 123 across 3 task formulations. For fair comparison, we use vanilla BERT-base as the base model for both SEED and FT methods.

	binary FEVER	FEVER	SciFact_oracle
p value	$4e^{-38}$	$1e^{-110}$	0.00679

Table 4.1: Statistical significance test results in 20-shot setting.

Table 4.1 presents p values. The p values are always smaller than 0.005, indicating statistical significance for performance improvements obtained by SEED across three task formulations. Noticeably the p value calculated on binary FEVER and three-way FEVER are much smaller than the p value on SciFact_oracle, which suggests that the performance advantages are less significant. It correlates well with task difficulty: SciFact is more challenging than FEVER. Overall, SEED achieves significant improvements over FT in 20-shot setting.

4.4 Analysis and Discussion

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 4.8 presents the standard deviation distribution on Binary FEVER claim verification, which is largely representative of the standard deviations of the models across the different settings. We only analyse configurations that utilise BERT-base and BERT-large for direct comparison

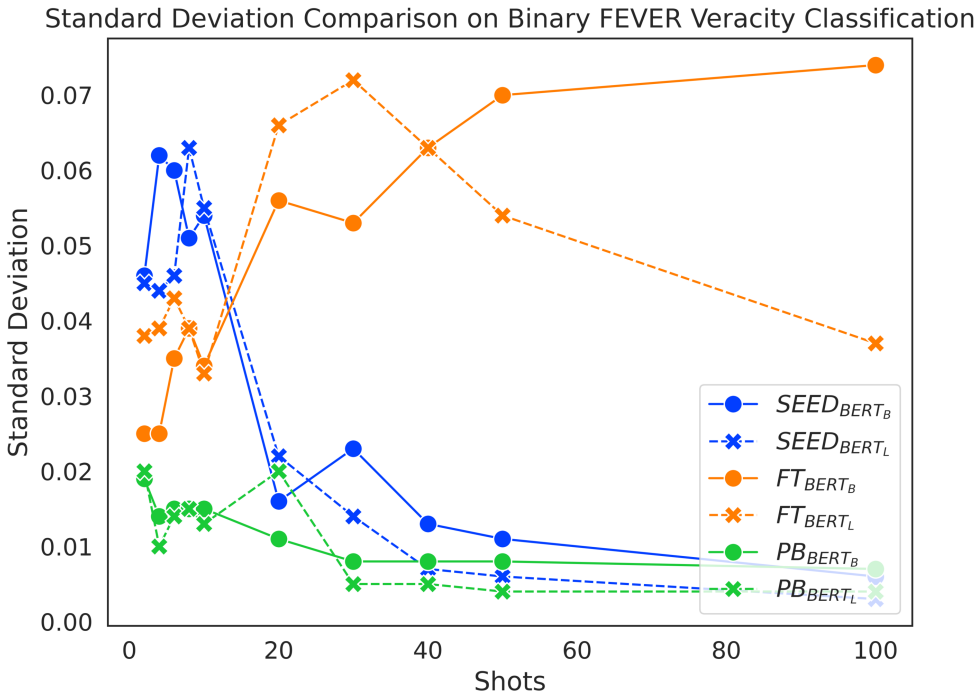

Figure 4.8: Standard deviation comparison on binary FEVER claim verification.

across methods. Overall, PB always has the lowest standard deviation, which demonstrates its low performance variability across random sampling seeds. Combined with the initial performance boost of SEED in Figure 4.2, the high standard deviation in the beginning implies that the SEED method is able to learn from the extremely limited number of training data and therefore experiences performance fluctuations due to different few-shot samples. Meanwhile, when given 10 or fewer shots, FT's accuracy performance remains close to random guess (see Figure 4.2) and its standard deviation remains low (see Figure 4.8). The low performance and the insensitivity to different sampling seeds indicates in this scenario that the FT method is not able to effectively learn from the extremely limited number of data. As the number of training samples further increases beyond 10 shots, the standard deviation of SEED drastically decreases and its performance experiences a boost until it converges at around 40 shots. After the initial performance boost, the SEED method shows robustness to random sampling. When given more than 10 shots, the standard deviations of FT surpass SEED with a large margin and its accuracy performance starts to experience a boost, which indicates that the FT models are able to learn from the given samples in this scenario. However, the FT models do not converge within the first 100 shots, which leads to high standard deviation within the range from 20-shots to 100-shots and they remain vulnerable

to random sampling in few-shot settings.

In short, PB is the most robust method to sample variations, despite underperforming SEED on average; SEED is still generally more robust to random sampling and has higher learning capacities than the FT method in few-shot settings.

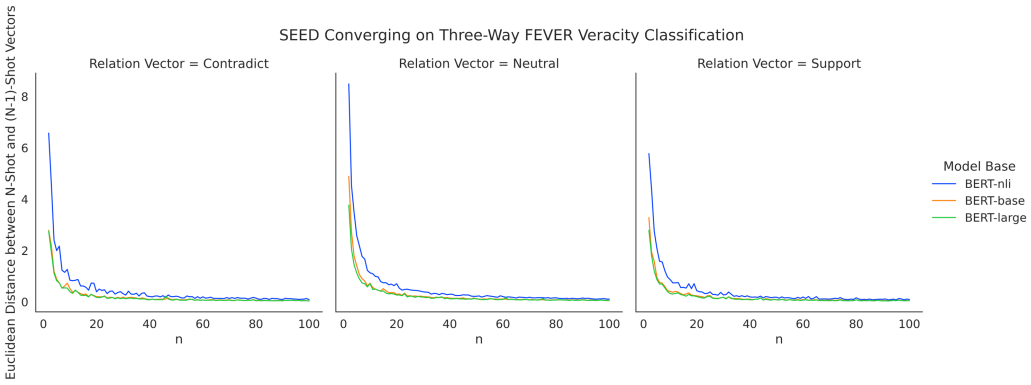

Figure 4.9: SEED converging on three-way FEVER claim verification with increasing number of *n* shots.

Why does SEED plateau? As presented in the Results Section, 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 100, 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 4.9 depicts the converging process with FEVER three-way claim verification. 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 with low demand on data quantity. It validates the use of semantic differences for verification and highlights its efficiency of data usage in few-shot settings. From a negative angle, it also means that the method stops learning as much for larger numbers of shots as it becomes stable, i.e. it is particularly useful in few-shot settings.

The curves of BERT-base and BERT-large largely overlap with each other, while the curve of BERT-nli does not conjoin until convergence. It corresponds well with the overall performance advantages of utilising BERT-nli as presented in the Results Section. 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 claim verification to address SciFact_oracle claim verification.

General discussions 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 claim verification, 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 claim verification 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 claim verification; 23% higher than FT on FEVER three-way claim verification and 17% higher than FT on SciFact_oracle three-way claim verification. Further, detailed analysis on classwise F1 performance also shows that improved performance is consistent across classes.

Our experiments successfully demonstrate that sentence embeddings from pre-trained language models can be effectively utilised to compute pairwise semantic differences between claims and their associated evidences with limited labelled instances. The proposed method leads to positive contributions with improved performance on the task of claim verification in few-shot settings. In comparison with PB, SEED has better learning capacities, higher few-shot performance, and most importantly, it is more flexible for doing multi-way claim verification, 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.

The main application scenario of SEED is few-shot pairwise classification, i.e. when the input involves text pairs. While we have demonstrated its effectiveness on few-shot claim verification, future work may study the effectiveness of SEED on other pairwise classification tasks, e.g., natural language inference, stance detection, knowledge graph completion and semantic relation classification between documents. Furthermore, SEED also offers the potential to be used for annotation quality evaluation: SEED is sensitive to data sampling within 10 shots and it may be utilised as a good metric to determine whether the annotated data is of high quality or not with

only a few samples. Moreover, SEED can be applied to do task difficulty estimation: SEED's few-shot performance on SciFact_oracle is significantly lower than FEVER, which correlates well with the fact the SciFact is more challenging than FEVER. In future studies, one may conduct few-shot experiments without gradient update using SEED on a new dataset and a familiar dataset to gain valuable initial understanding on the difficulty of the new dataset.

While SEED demonstrates the ability to learn representative vectors that lead to effective claim verification with limited labelled data and computational resources, its design remains simple and its performance plateaus with larger numbers of shots. Future studies may further develop the method by utilising more advanced sentence embeddings. For example, while our proposed SEED calculates mean values of all tokens for sentence embeddings, future work may obtain syntactically aware sentence embeddings by calculating weighted average values with reference to syntactic parse trees. In addition, further exploration into SEED's potential to further improve its performance when more training samples are observed would also be a valuable avenue of future research. One possibility to achieve this could be by extending SEED with the use of gradient descent.

4.5 Summary

We have presented an efficient and effective SEED method which achieves significant improvements over the baseline systems in few-shot claim verification. By comparing it with a perplexity-based few-shot claim verification 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 applied, for example, to new domains with limited labelled examples. Future research may further extend SEED with more sophisticated sentence embeddings. While our focus here has been on few-shot learning, future research could focus on building a capacity to more effective learning from larger numbers of training samples.

Chapter 5

MAPLE: Micro Analysis of Pairwise Language Evolution for Few-Shot Claim Verification

In last chapter, we presented SEED, an effective claim verification method that does not perform gradient descent to update model parameters for high efficiency and works particularlly well when combined with an NLI-trained PLM. However, its best performing configuration heavily relies on NLI-trained PLMs, limiting its applicability to only cases where NLI data and/or NLI-trained PLMs are available, excluding scenarios such as low-resource languages. Moreover, it excels when the data is similar to general NLI data but struggles when dealing with dissimilar data, such as claim verification data where the evidence is particularly long and technical. In this chapter, we propose to embrace the potential of performing some more computing with gradient descent and leveraging unlabeled fact-checking data rather than general NLI data, to enhance few-shot claim verification.

We present MAPLE (Micro Analysis of Pairwise Language Evolution), a novel approach designed for few-shot claim verification. MAPLE innovatively builds upon the concept of language transition[1], scrutinizing the semantic shift that occurs as a sequence-to-sequence model learns to generate a target sequence from a given input sequence. In this chapter, such language transition

[1]In this chapter, we distinguish between claim language and evidence language, treating them as distinct languages as they may differ in formality, length, or even depth. In real-world scenarios, checkworthy claims often emanate from more informal settings, such as social media platforms. On the other hand, evidences typically come from formal and reputable sources such as research papers and Wikipedia, marked by a concise, informative, and professional style. For concrete examples, please see the data samples in Section 3.1.2.

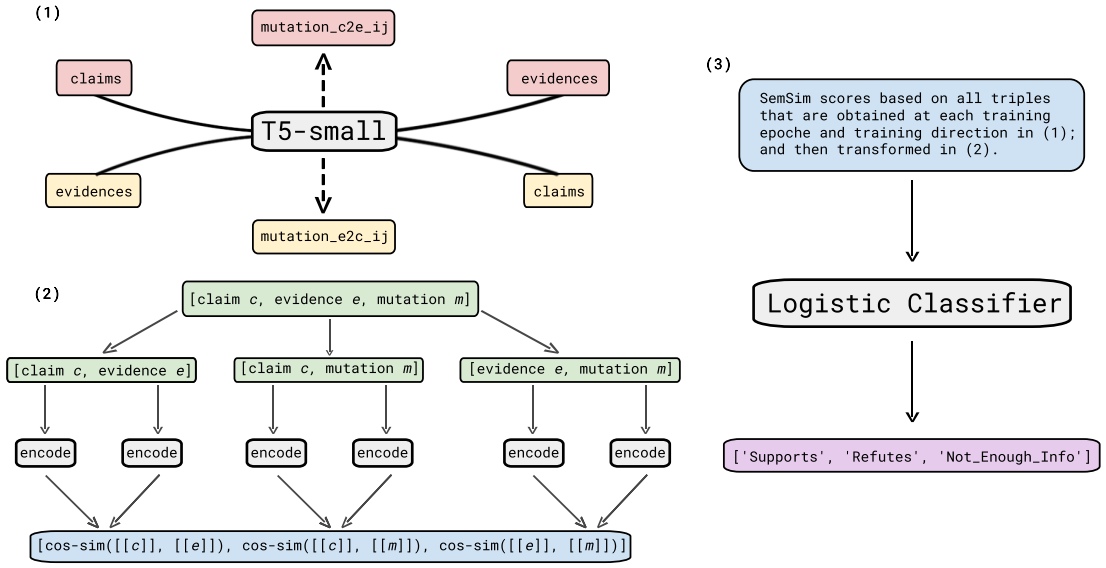

Figure 5.1: MAPLE for claim verification. **(1) In-domain seq2seq training.** With LoRA, a T5-small model is trained on claim-to-evidence task for e epochs using the d unlabelled claim-evidence pairs from the data pool. At the end of each training epoch j, model inference is performed on each instance i to generate a mutation *mutation_c2e_i*. This process is repeated on evidence-to-claim setting. In total this step produces $2 * d * e$ triples that consist of a claim c, an associated piece of evidence e and a generated mutation m. **(2) SemSim transformation.** Each triple is grouped into three pairs including claim-evidence pair $c-e$, claim-mutation pair $c-m$ and evidence-mutation pair $e-m$. 'Semsim' scores are obtained for each pair by calculating the cosine similarity score based on corresponding sentence embeddings. **(3) Logistic classifier training with few-shot labelled data.** A logistic classifier is trained on labelled data where the transformed 'SemSim' scores are used as input features to predict veracity labels.

from the input sequence to the output sequence over the training epochs is referred to as pairwise language evolution. As the semantic similarity within a text pair can be reflected by how difficult it is for a sequence-to-sequence model to learn from and converge upon, MAPLE is designed to capture such signals and use them as input features to make predictions for claim verification task. By intricately capturing and harnessing this pairwise language evolution, MAPLE aims to facilitate accurate predictions even in scenarios with minimal labeled data. Our key novel contributions include:

- We introduce MAPLE, an innovative approach that leverages unlabeled data for enhancing few-shot claim verification. While building MAPLE, we also propose 'SemSim' as an NLG evaluation metric that focuses on semantic similarity.

- We perform a pioneering exploration of the language transition convergence process during seq2seq model training.

- We conduct comprehensive experiments on four dataset configurations, facilitating a direct comparison with established SOTA methods, namely SEED, PET, and LLaMA 2.

Experiments demonstrates MAPLE's effectiveness and robustness across different dataset domains and configurations, when unlabeled data for seq2seq in-domain training is available, particularly within five shots. Fact-checking practitioners may leverage MAPLE to perform few-shot claim verification when unlabeled data and some computational resources are available, while labeled data is extremely limited.

5.1 Methodology

Traditionally, generative models are often used in classification tasks by generating corresponding labels given input sentences (Pradeep et al., 2021). However, such an approach does not fully exploit the potential of generative models on tasks such as claim verification. In this section, we present the MAPLE method and its application to few-shot claim verification.

The intuition of MAPLE is that sentence pairs of various relationships bring diverse learning challenges to the seq2seq generation task. As the data difficulty is reflected in the seq2seq training process, such learning difficulty associated with each sample could be further transformed into various signs to indicate the relationship within a sentence pair. We explore such potential to be leveraged for effective claim verification, where the goal is to determine the veracity of a claim based on its relationship with the provided evidence. MAPLE consists of three steps, as illustrated in Figure 5.1.

(1) In-domain seq2seq training. In order to leverage in-domain unlabeled data, i.e. claim-evidence pairs without veracity labels, we perform seq2seq training in two directions: claim-to-evidence and evidence-to-claim. For claim-to-evidence task, a T5-small Raffel et al. (2020) model is fine-tuned for e epochs using all of the unlabeled claim-evidence pairs from the data pool with a size of d. At the end of each training epoch j, model inference is performed on each instance i to generate a mutation $mutation\_c2e\_i$. Similarly, another T5-small model is fine-tuned on evidence-to-claim task to generate mutations $mutation\_e2c\_i$ for each training epoch j. For computational efficiency, the training is conducted with Low-Rank Adaptation (LoRA) [2] Hu et al. (2022a), a parameter-efficient training method. In total, this step produces $2 * d * e$ triples that consist of a claim c, an associated piece of evidence e and a generated mutation m.

[2]Please see more information on the training algorithms in section 5.4.

(2) SemSim transformation. The SemSim transformation aims to transform the generated triples into numeric scores while recording the transition of mutation m during the training process in both claim-to-evidence task and evidence-to-claim task. Each triple is grouped into three pairs including claim-evidence pair $c-e$, claim-mutation pair $c-m$ and evidence-mutation pair $e-m$. We measure the pairwise similarity with 'SemSim' score: first obtains sentence embeddings with model 'sentence-transformers/all-mpnet-base-v2' Reimers and Gurevych (2019), a sentence transformer model that is trained on over one billion sentences with contrastive training objective; then calculates cosine similarity scores on sentence embeddings for each pair. Each triple is transformed into an array of 3 'SemSim' scores. All triples of a claim-evidence instance are concatenated as features of the instance.

(3) Logistic classifier training with few-shot labeled data. Using n-shot labeled data from the labeled data pool of size $3n$,[3] i.e. claim-evidence pairs with veracity labels, a logistic classifier is trained. The transformed SemSim scores are used as input features to make predictions on veracity labels.[4]

5.2 Experimental settings

In this section, experiments comparing MAPLE with previous SOTA methods on four dataset configurations as presented in 3.1.

Baselines **SEED** SEED uses a sentence-transformer model that is trained on NLI tasks.[5]

PET PET uses BERT-base fine-tuned on the MNLI dataset.[6] It is trained with a batch size of 16, a learning rate of $1e^{-5}$, and training epochs of 3, following previous practice (Schick and Schütze, 2021a,b; Zeng and Zubiaga, 2023).

LLaMA 2 LLaMA 2 experiments are conducted on the LLaMA 2 7b chat model.[7] Answers

[3]For example, 1-shot experiments are conducted on a data pool that includes 3 labeled samples in total, i.e., one instance per class per claim verification task.

[4]Please note that MAPLE differs from data augmentation methods. Data argumentation generates pseudo-data and uses them as additional samples for model training; MAPLE does not treat mutations as additional training samples, but relies on them to obtain input features for logistic classifier training. From a tabular view, typical data augmentation methods generate additional rows but MAPLE operates on columns.

[5]Huggingface hub model id 'bert-base-nli-mean-tokens' (Zeng and Zubiaga, 2022).

[6]Huggingface hub model id 'textattack/bert-base-uncased-MNLI'. See performance using alternative model checkpoint in Appendix C.1.

[7]Huggingface hub model id 'Llama-2-7b-chat-hf'. See performance using alternative model checkpoint in Appendix C.1.

are generated by prompting with detailed instructions[8] and post-processed to match class labels [9].

MAPLE In our experiments, MAPLE uses the T5-small model for efficient training.[10] Training is conducted with LoRA from epoch 0 to epoch 20, using 0.0001 as learning rate, 16 as batch size, 512 as max length, 0.1 as LoRA dropout, 32 as LoRA alpha Hu et al. (2022a) and "Summarize:" as the prompt (Ramamurthy et al., 2022).

Experimental Setup Our experimental setup is designed to conduct comprehensive few-shot experiments, where the term 'n-shot' refers to the number of samples available per class. As we focus on few-shot performance, our main experiments are conducted on 1-shot, 2-shot, 3-shot, 4-shot and 5-shot settings. To ensure the reliability and generalizability of our findings, each n-shot experiment has been repeated 100 times with sampling seeds ranging from 123 to 223. We present the main results in Section 5.3. We also present further experiments showing the trend going up to 50 shots in Appendix C.3.

Figure 5.2: F1 performance within 5 shots.

5.3 Results

In this section, we present the results of our experiments with a focus on few-shot settings.

Figure 5.2 illustrates the F1 performance within the 5-shot setting.[11] Across the four dataset configurations, MAPLE shows noticeable performance advantages within the 5-shot setting, validating its effectiveness in few-shot scenarios and robustness across datasets. It achieves this

[8] After evaluating several prompts, the subsequent one is employed due to its superior performance.: "Please perform the task of claim verification: you are given a claim and a piece of evidence, your goal is to classify the pair out of 'SUPPORTS', 'REFUTES' and 'NOT_ENOUGH_INFO'. Here are a few examples: claim: train_claim_i evidence: train_evidences_i label: train_label_i What is the label for the following pair out of 'SUPPORTS', 'REFUTES' and 'NOT_ENOUGH_INFO'? Answer with the label only. "

[9] Post-processing primarily includes stripping formatting strings and removing "label: ". The remaining responses that do not belong to any of the labels are mapped into the "NOT_ENOUGH_INFO" class, e.g. responses such as "?" and "Please give me the answer".

[10] Huggingface hub model id 't5-small' Raffel et al. (2020).

[11] Please see detailed classwise performance in Appendix C.2

primarily by starting from a high performance point and steadily improving within 5 shots. Although SEED underperforms MAPLE, it showcases strong learning capabilities, and its relatively lower performance is primarily due to a low starting point. Surprisingly, PET and LLaMA 2 perform poorly within the 5-shot range, generally starting low and exhibiting limited learning capabilities.

On the FEVER dataset, MAPLE demonstrates significant improvements over the baselines. Specifically, MAPLE achieves a very high F1 score over 0.6 at 1 shot, outperforming SEED, PET, and LLaMA 2, which commence at approximately 0.25, 0.37, and 0.38, respectively. Within 5 shots, MAPLE exhibits a steady performance improvement, surpassing an F1 score of 0.7. While SEED and PET also experience notable performance boosts, with SEED approaching just below 0.6 and PET reaching below 0.5, LLaMA 2 encounters a slight performance drop, settling around 0.36.

On the cFEVER dataset, the performance of all methods exhibits a considerable decrease compared to FEVER, highlighting the challenging nature of the dataset. While MAPLE maintains its leading position overall, the performance margin is narrower. It initiates above 0.3 and achieves scores surpassing 0.4. SEED begins even lower, below 0.3, but manages to surpass 0.4, albeit slightly trailing behind MAPLE. PET encounters greater challenges overall, commencing below SEED and only slightly exceeding 0.3. LLaMA 2 excels initially with a score of 0.38 but experiences a drop to 0.37.

On the SciFact_oracle dataset configuration, despite the overall performance being better than cFEVER but worse than FEVER across all methods, MAPLE maintains superiority within 5 shots. It initiates around 0.4 and concludes around 0.45. SEED begins around 0.3 and lags behind MAPLE, while PET starts higher than SEED but lower than MAPLE, failing to surpass them within 5 shots. LLaMA 2 performs comparably to PET, starting at 0.37 and finishing at 0.40.

On the SciFact_retrieved dataset configuration, MAPLE demonstrates a slightly better performance compared to SciFact_oracle, while all baseline methods exhibit a substantial decline in performance compared to SciFact_oracle. Consequently, MAPLE achieves a larger performance margin. It commences above 0.4 and concludes around 0.5. SEED starts at a very low point, below 0.3, and approaches 0.4 at 5 shots. PET initiates around 0.35 but struggles to learn effectively within 5 shots, resulting in an even lower score. LLaMA 2 starts at 0.32 and 0.29 and experiences a notable drop to 0.18 and 0.17 immediately afterwards.[12]

[12]Note that the SciFact_retrieved dataset configuration comprises lengthy instances that may exceed the

In general, LLaMA 2 displays reasonable one-shot performance but shows limited learning capabilities within 5 shots. Despite PET's use of gradient descent to update the parameters of a large language model, this strategy does not yield satisfactory results within the 5-shot range. On the other hand, MAPLE and SEED showcase relatively rapid convergence due to their limited number of trainable parameters. MAPLE stands out with a significantly higher level of performance compared to all baselines overall, demonstrating its capacity to leverage limited data for notable results and effectiveness as a few-shot claim verification model.

It's crucial to highlight that while most experiments are conducted in oracle settings, real-world claim verification often introduces the challenge of imperfect evidences. Therefore, achieving optimal performance in the SciFact_retrieved dataset, where evidence is noisy and lengthy, is particularly significant. This accomplishment highlights MAPLE's robustness to noisy and challenging data in realistic fact-checking scenarios.

5.4 Ablation Studies

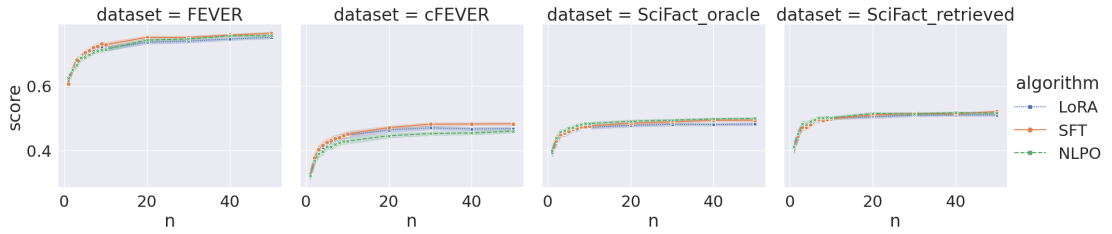

Figure 5.3: Comparison of MAPLE performance using different training algorithms for in-domain seq2seq training. The label "LoRA" represents parameter-efficient training method Low-Rank Adaptation, "SFT" indicates supervised fine-tuning and "NLPO" refers to reinforcement learning with the NLPO policy.

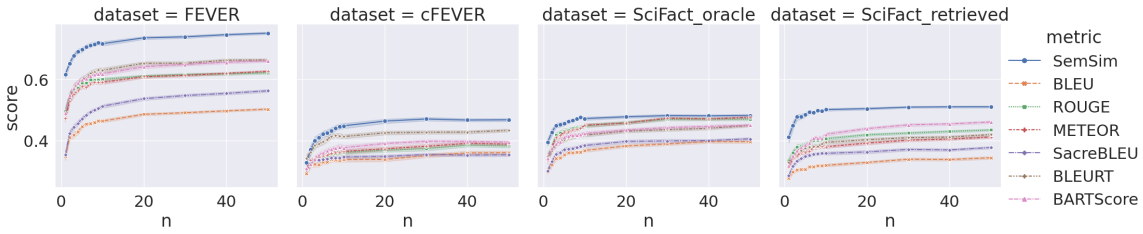

Figure 5.4: Comparison of MAPLE performance using the proposed 'SemSim' metric and alternative metrics to measure micro pairwise language evolution.

Training algorithms With the growing interest in reinforcement learning (RL) and parameter-efficient training, this ablation study investigates the effects of utilizing different training algo-

maximum context length for LLaMA 2. Addressing this issue would necessitate additional techniques.

rithms. Specifically, we comprare LoRA, Supervised Fine-Tuning (SFT) and Natural Language Policy Optimization (NLPO) and compare MAPLE results on few-shot claim verification when seq2seq in-domain training is performed with these different training algorithms.

SFT is the canonical task adaptation training algorithm. For a model that has a weight matrix W, it computes a weight update matrix ΔW during backpropagation which has the same size as matrix W and contains the information for the model to update in order to minimise the loss function. Compared with SFT, LoRA (Hu et al., 2022a) is a paramter-efficient training algorithm that reduces the number of trainable parameters without introducing inference latency. It achieves an approximate of SFT by freezing the original model weights W and replace the full size $d_1 * d_2$ matrix ΔW with the decomposition of ΔW: two smaller LoRA matrices, A $(d_1 * r)$ and B $(r * d_2)$, where r is a new hyperparameter r that is significantly smaller than d_1 and/or d_2. With vastly reduced computing storage requirement and computing time, experiments show that LoRA achieves similar performance results. NLPO (Ramamurthy et al., 2022) is a novel on-policy RL algorithm that dynamically learns task-specific constraints over the distribution of language at a token level. As natural language generation can be viewed as a sequential decision making processing, there is growing interest in applying RL training algorithms to PLMs. Since language generation action spaces are significantly larger than traditional applications of RL algorithms, it could cause instability when training PLMs with traditional RL methods. NLPO introduces top-p sampling to mask out less relevant tokens in-context as it trains. It offers enhanced stability and performance compared to previous policy gradient methods (Ramamurthy et al., 2022).

As presented in Figure 5.4, the overall differences in performance among the algorithms are relatively marginal. SFT demonstrates best results on the FEVER and cFEVER datasets, while NLPO outperforms on the SciFact_oracle and SciFact_retrieved datasets. Notably, despite the largely reduced computational burden by utilizing LoRA,[13] the observed performance drops are modest. Therefore, MAPLE conducts in-domain seq2seq training with LoRA.

Metrics MAPLE uses our proposed 'SemSim' metric to measure and analyze the pairwise language evolution. This ablation section presents the comparison with a number of established NLG metrics, including 'BLEU', 'ROUGE', 'METEOR', 'SacreBLEU', 'BLEURT', and 'BARTScore'.

Figure 5.4 illustrates the performance variations of MAPLE when employing different metrics. Across all datasets, the 'SemSim' metric demonstrates superior performance compared to other

[13]For T5-small, the trainable % with LoRA is 0.485 (294,912/60,801,536). Please see a detailed efficiency comparison with SFT in Appendix D.1.

metrics, showcasing a significant improvement gap. This highlights the advantages of 'SemSim', establishing it as the optimal choice for MAPLE. By focusing on measuring semantic similarity as a primary component, we can effectively analyze the micro pairwise evolution of language in a seq2seq learning process, which is captured by generated mutations across training epochs. In contrast, metrics based solely on lexical overlap, or utilizing an LLM that is not trained on substantial sentence pair data, may be less indicative in capturing the nuances of language evolution. The emphasis on fine-grained semantic similarity provides highly informative insights, particularly in assessing the learning difficulty of instances for seq2seq generation. As 'SemSim' surpasses many established NLG metrics in this task, it shows its potential for broader applications as a general NLG evaluation metric.

5.5 Analysis and Discussion

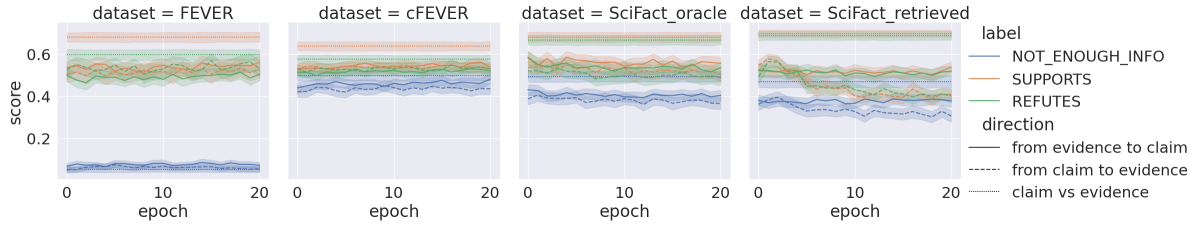

Figure 5.5: Example signals captured for classification, using the 'SemSim' score for target-mutation pairs on the test.

Despite recent research on generating rationales and explanations Atanasova et al. (2020); Kotonya and Toni (2020); Schuster et al. (2021), existing approaches heavily depend on directly fine-tuning PLMs, hindering the understanding of their decision-making process. MAPLE stands out by providing tangible and traceable solutions, guided by the principle that sentence pairs with different relations present distinct challenges for seq2seq generation. Figure 5.5 further supports this principle and elucidates the effectiveness of MAPLE. Overall, the 'SemSim' scores for 'NOT_ENOUGH_INFO' are significantly lower than those for 'SUPPORTS' and 'REFUTES', enabling easy differentiation between 'NOT_ENOUGH_INFO' and other classes [14]. Furthermore, generating a piece of evidence from a claim proves to be more challenging than generating a claim from a piece of evidence. Generating claims primarily needs the removal of redundant or unnecessary content, while generating evidence requires the model to expand the existing

[14]The detailed classwise performance in Appendix C.2 shows that MAPLE has the best performance on 'NOT_ENOUGH_INFO' class.

content. Furthermore, figure 5.5 shows that generating a claim is easier for 'SUPPORTS' than for 'REFUTES', while generating evidence is easier for 'REFUTES' than for 'SUPPORTS'. This pattern allows for a distinction between the two categories. With its enhanced interpretability and traceability, MAPLE aims to bolster the reliability and trustworthiness of the claim verification process.

Moreover, by comparing the difficulty among datasets based on the above information, we can gain insights into the varying challenges posed by different domains. For example, if a dataset such as FEVER consistently exhibits high 'SemSim' scores and low standard deviation during in-domain seq2seq training, it suggests that the claims and evidences within that dataset are easier to match and converge upon. On the other hand, datasets such as cFEVER with lower 'SemSim' scores, higher standard deviation, and longer convergence time indicate greater difficulty in aligning claims and evidences. This comparative analysis allows us to understand the relative complexities of fact-checking in different settings and further enhances the interpretability of MAPLE's performance across datasets.

Moreover, MAPLE's low demand on annotations and computing facilities enhances its efficiency and accessibility. Both step (1) in-domain seq2seq training and step (2) SemSim transformation only require unlabeled claim-evidence pairs and limited annotations are only required for step (3) logistic classifier training with few-shot labelled data. While performing steps (1) and (2) over the entire unlabeled pool may seem burdensome, such practice only takes from minutes to few hours.[15] Due to MAPLE's efficiency and accessibility by design, training and deploying can be easily accomplished on Google Colab with a free account or even on a personal laptop. In real-world scenarios where the claim verification team has accumulated a substantial collection of claim-evidence pairs, which can be claims with annotated oracle evidences or claims with retrieved noisy evidences, they can initiate steps (1) and (2) and this process can be completed while the team actively acquires a small number of labeled samples. Subsequently, step (3) training a logistic classifier with the newly acquired data only takes seconds and MAPLE is ready for deployment. By designing such an efficient workflow, the application of MAPLE in real-world scenarios can bring in a decent claim verification model with minimal cost in annotation and computational resources. Overall, MAPLE holds practical value for fact-checking in real-world contexts, particularly as a tool to assist fact-checkers in combating emerging domains

[15] Please see detailed overall runtime report in Appendix D.2.

of misinformation.

Future Directions With the development of MAPLE, several promising directions for future research emerge:

Self-supervised Extensions Currently, MAPLE combines language transition signals with a traditional logistic classifier for classification. A further research avenue could include its development into a fully self-supervised system by integrating clustering methods.

NLG metric Adaptability While we propose 'SemSim' as an NLG metric and have demonstrated its performance advantages for MAPLE, a comprehensive evaluation of 'SemSim' for broader tasks and domains would enhance the understanding.

Most prevalent NLG evaluation metrics currently calculate similarity scores based on sentence embeddings only, including the proposed metric 'SemSim' in this chapter, whereas MAPLE offers nuanced insights derived from the seq2seq training dynamics. Converting MAPLE, which combines 'SemSim' and T5 training, into a general NLG evaluation metric would be a promising research direction.

Human-in-the-loop Workflow As previously demonstrated, MAPLE shows potential for assisting fact-checkers in real-world scenarios. Fully exploring this potential primarily involves leveraging MAPLE as a claim verification model in fact-checking organizations. Additionally, it can serve as the backbone of an active learning system, facilitating data annotation prioritization.

5.6 Summary

In this chapter, we have introduced MAPLE, a novel approach for few-shot claim verification. By leveraging language transition signals during seq2seq training convergence, MAPLE achieves SOTA performance in precisely predicting claim veracity labels with reference to associated evidences in few-shot learning scenarios. Through extensive experiments and analysis on multiple datasets, we validate its effectiveness, robustness, interpretability, efficiency and accesibility.

Chapter 6

Active PETs: Active Data Annotation Prioritisation for Few-Shot Claim Verification with Pattern Exploiting Training

---

Where new domains needing fact-checking emerge, collecting and annotating labelled data can carry an impractical delay. Hence, we focus on few-shot claim verification task and have presented two few-shot claim verification methods: SEED in chapter 4 and MAPLE in chapter 5, both of which perform experiments using few-shot data constructed from random sampling. However, given the cost and effort of labelling fact-checking data, practitioners can often be selective in labelling a small subset, particularly when the availability of unlabelled data is abundant. In these circumstances, rather than randomly sampling this subset, we propose to optimise the selection of candidate instances to be labelled through active learning, such that it leads to overall improved few-shot performance.

In this chapter, we represent the first such effort in proposing an approach leveraging an active learning strategy for the claim verification problem to study how to optimise the usage of a highly constrained annotation budget, as well as the first in furthering Pattern Exploiting Training (PET) with an active learning strategy. To achieve this, we propose Active PETs, a novel methodology that enables the ability to leverage an active learning strategy through a committee of PETs. Figure 6.1 illustrates the application of the active learning strategy on data annotation priotisation. For each iteration, firstly the committee retrieves k new unlabelled samples (k=10 in our experiments),

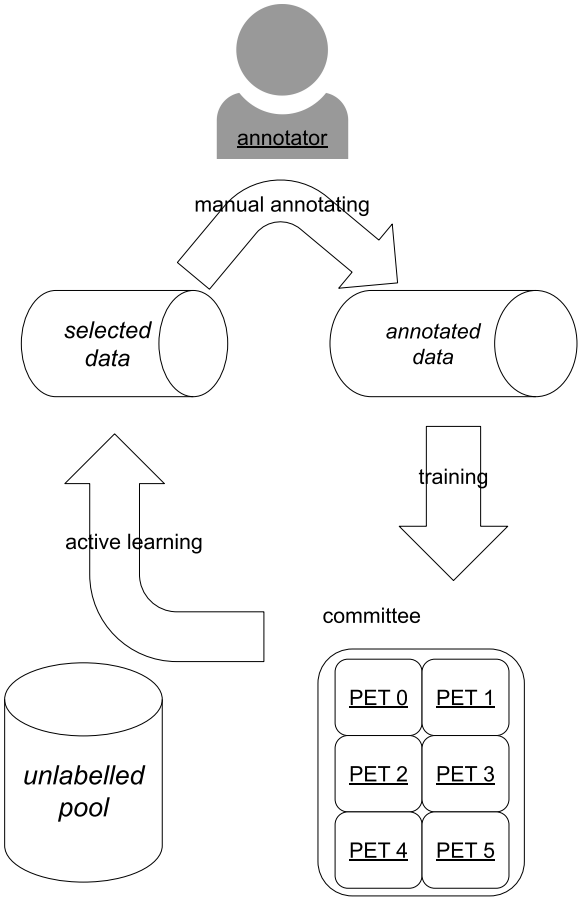

Figure 6.1: Illustration of the data annotation prioritisation scenario with a committee of 6 PETs.

secondly the human annotators label them, lastly each of the PET based on different PLMs is trained individually with all of the labelled samples at hand. Our experiments start from 0 labelled samples and end at 300 labelled samples.

By exploring effective prioritisation of unlabelled data for annotation and making better use of a small amount of labelled data, we make the following novel contributions:

- we are the first to study data annotation prioritisation through active learning for few-shot claim verification;
- we are the first to study the extensibility of PET to enable active learning, by proposing Active PETs, a novel ensemble-based cold-start active learning strategy that enables multiple pretrained language models (PLMs) to collectively prioritise data instances;
- we further investigate the effect of oversampling on mitigating the impact of imbalanced data selection on few-shot learning, when guided by active learning;

- we conduct further corpus-based analysis on the selected few-shot data instances, which highlights the potential of Active PETs to lead to improved lexical and semantic characteristics that benefit the task.

Our results show consistently improved performance of Active PETs over the baseline active learning strategies on two datasets, SciFact (Wadden et al., 2020) and Climate FEVER (Diggelmann et al., 2021). In addition to improved performance over the baselines, our research emphasises the importance of the hitherto unexplored data prioritisation in claim verification, showing remarkable performance improvements where time and budget are limited.

Active PETs achieve significant overall improvements for few-shot claim verification with highly constrained annotation budget, particularly when the unlabeled pool exhibits heavily skewed data distribution for three-way claim verification, a common scenario in real-world fact-checking. Following the diagram in Figure 6.1, practioners can leverage Active PETs to actively select candidate samples from the unlabelled pool, obtain the annotations and train models accordingly for optimise overall performance.

6.1 Methodology

In this section, we introduce our model Active PETs, and describe the oversampling mechanism we use.

Proposed method: Active PETs

Having a large pool of unlabelled data, our objective is to design a query strategy that selects suitable candidates to be labelled, such that the labelled pool of instances leads to optimal few-shot performance. Our query strategy is rooted in the intuition that disagreement among different PETs in a committee can capture the uncertainty of a particular instance.

Based on the assumption that performance of different language models is largely dependent on model size (Kaplan et al., 2020), we introduce a weighting mechanism: each PET is first assigned a number of votes V_i that is proportional to its hidden size,[1] and ultimately all votes are aggregated. Algorithm 1 presents the pseudo-code for executing a single query iteration with Active PETs.

We then quantify the disagreement by calculating vote entropy (Dagan and Engelson, 1995):

[1]For example, if we use a committee formed of only base models that have 6 hidden layers and large models that have 12 hidden layers, proportionally each of the base models is allocated one vote and each of the large models is allocated two votes.

Algorithm 1 A Single Query Iteration

Require: The last trained Commitee of PETs C, unlabelled data pool U, query size k

for $PET_i \in C$ **do**

$v_i \leftarrow Size(PET_i) / \min_{\forall PET_i \in C} Size(PET_i)$

end for ▷ assign number of votes

for instance $x \in U$ **do**

for $PET_i \in C$ **do**

$V_{x_i} \leftarrow resize(\hat{y}_{x_i}, v_i)$

end for ▷ predict label and vote

$S_x = -\sum_{\forall V_{x_i} \in V_x} \frac{V_{x_i}}{|V|} log \frac{(V_{x_i})}{|V|}$

end for ▷ calculate entropy scores

return $Sort(S)[:k]$ ▷ return top k instances

$$score_x = -\sum_{\hat{y}} \frac{vote(x,\hat{y})}{count(V)} \log \frac{vote(x,\hat{y})}{count(V)} \tag{6.1}$$

where $\hat{y}$ is the predicted label, x is the instance, $vote(x,\hat{y})$ are the committee votes of $\hat{y}$ for the instance x, and $count(V)$ is the number of total assigned votes. It can be viewed as a QBC generalisation of entropy-based uncertainty sampling that is designed to combine models of different sizes.

Data Oversampling

One of the risks of the proposed active learning strategy is that the resulting training data may not be adequately balanced, which can impact model performance. An accessible solution is oversampling: resample the instances from the minority class with replacement until balanced. Note that this does not increase the labelling effort as instances are repeated from the labelled pool. Instead of random resampling (Japkowicz, 2000), we propose a novel technique of integrating resampling with the committee's preference. For each minority class, we start resampling from the instance that has the highest disagreement score to the instance that has the lower disagreement score. In highly imbalanced cases, resampling is repeated from the highest to lowest priority until the overall label distribution is balanced. Algorithm 2 presents the pseudo-code for executing the training loop with the option of conducting oversampling with Active PETs.

Algorithm 2 Training

Require: Labelled and sorted data D, A initial Commitee of PETs C

if Oversampling **then**

$c \leftarrow max_{\forall class \in D} count(data \in class)$

$D \leftarrow resize_{\forall class \in D}(class, c)$

end if ▷ oversampling

for $PET_i \in C$ **do**

$PET_i \leftarrow train(PET_i, D)$

end for ▷ train the commitee of PETs

return C ▷ return trained PETs

6.2 Experimental Settings

	SciFact_retrieved		
	'SUPPORTS'	**'NOT_ENOUGH_INFO'**	**'REFUTES'**
UP	266 (9.31%)	2530 (88.55%)	61 (2.14%)
Test	150 (33.33%)	150 (33.33%)	150 (33.33%)
	cFEVER		
	'SUPPORTS'	**'NOT_ENOUGH_INFO'**	**'REFUTES'**
UP	1789 (24.78%)	4778 (66.19%)	652 (8.66%)
Test	150 (33.33%)	150 (33.33%)	150 (33.33%)

Table 6.1: Label distribution of SciFact_retrieved and cFEVER. UP = unlabelled pool of training data.

The main experiments focus on datasets with real-world claims in a realistic setting: SciFact_retrieved dataset configuration and cFEVER dataset, known to be challenging, technical and free of highly synthetic data. Their reformulated data is highly imbalanced as presented in Table 6.1.

Baselines We compare our method to four baselines: random sampling, BADGE, CAL and ALPS.

Random For random sampling, we run each experiment over 10 different sampling seeds ranging from 123 to 132, and present the averaged results.

BADGE BADGE (Ash et al., 2020) optimises for both uncertainty and diversity. Gradient embeddings g_x are first computed for each data in the unlabelled pool, where g_x is the gradient of the cross entropy loss with respect to the parameters of the model's last layer. It then applies k-MEANS++ clustering on the obtained gradient embeddings, and batch selects instances that differ in feature representation and predictive uncertainty.

Though BADGE is proposed as a warm-start method, the required initial set of labelled data is only used for the initial training the model. In our experiments on claim verification, PLMs that are already finetuned on a similar task NLI are used, hence, BADGE can be used for cold-start sampling.

CAL CAL (Margatina et al., 2021), the SOTA warm-start strategy, highlights contrastive data points: data that has similar model encodings but different model predictions. Unlike BADGE, an initial labelled set of data is essential for CAL. It first calculates the [CLS] embeddings for all of the data and then runs K-Nearest-Neighbours (KNN) to obtain the k closest labelled neighbours for each unlabelled instance. It further calculates predictive probabilities from the model and measures Kullback-Leibler divergence on it. Finally it selects unlabelled instances whose predictive likelihoods diverge the most from their neighbours.

While CAL achieves SOTA performance as a warm-start strategy, its dependence on an initial labelled set of data makes it incompatible in the same few-shot active learning settings without an initial labelled set. However, for comprehensive comparison purposes, we still include it as a baseline starting at 100 labelled instances that are obtained from random sampling with 10 different random seeds.

ALPS ALPS (Yuan et al., 2020), the SOTA cold-start active learning method, also aims to take both model uncertainty and data diversity into account. It calculates surprisal embeddings to represent model uncertainty. Specifically, for each instance x, it is passed through the masked language modelling head of a PLM and then 15% of the tokens in x are randomly selected to calculate the cross entropy against their target tokens. The surprisal embeddings go through L2-normalisation and then get clustered to select the top samples.

Active PETs Committees of five to fifteen models are common for an ensemble-based active learning strategy (Settles, 2012). Here we form a committee of 6 PETs with 3 types of PLMs:

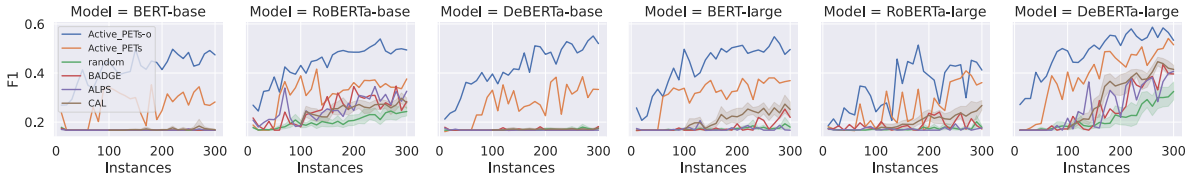

Figure 6.2: Few-Shot F1 Performance on SciFact_retrieved claim verification.

BERT-base, BERT-large (Devlin et al., 2019), RoBERTa-base, RoBERTa-large (Liu et al., 2019), DeBERTa-base and DeBERTa-large (He et al., 2021). Given the commonalities between the NLI and claim verification tasks, we use the PLM checkpoints already fine-tuned on MNLI (Williams et al., 2018).

Despite a line of research in optimising PET patterns and verbalisers (Tam et al., 2021), that is not our main focus. We use the following pattern and verbaliser for PET: [claim]? *[mask]*, [evidence]; 'SUPPORTS':"Yes", 'REFUTES':"No", 'NOT_ENOUGH_INFO':"Maybe", as they yielded best performance on NLI tasks in our preliminary experiments. Figure 6.3 provides an example of performing claim verification using PET.

There are two steps in our approach: (1) an ensemble method is used for data annotation prioritisation, after which data is selected and annotated, and (2) with the data instances drawn and annotated, we train a PET model that uses a single PLM to make the predictions. An ensemble method is key in step (1) to support the combined decision-making of choosing instances to annotate, but not in step (2) for the PET model which runs on a single PLM. Hence, results are presented for individual PETs, even if in all cases the ensemble is involved in the underlying prioritisation step. We test two variants: **Active_PETs** with no oversampling, and **Active_PETs-o** with the oversampling described in Section 6.1.

Experimental Setup **Hyperparameters.** As in few-shot settings we lack a development set, we follow previous work (Schick and Schütze, 2021a,b) and use the following hyperparameters for all experiments: $1e^{-5}$ as learning rate, 16 as batch size, 3 as the number of training epochs, 256 as the max sequence length. [2]

Labelling budget. We set it to a maximum of 300. We experiment with all scenarios ranging from 10 to 300 instances with a step size of 10.

Checkpoints. We always use the PLM checkpoints from the last iteration to perform active

[2]See further details for reproducibility in Appendix D.3.

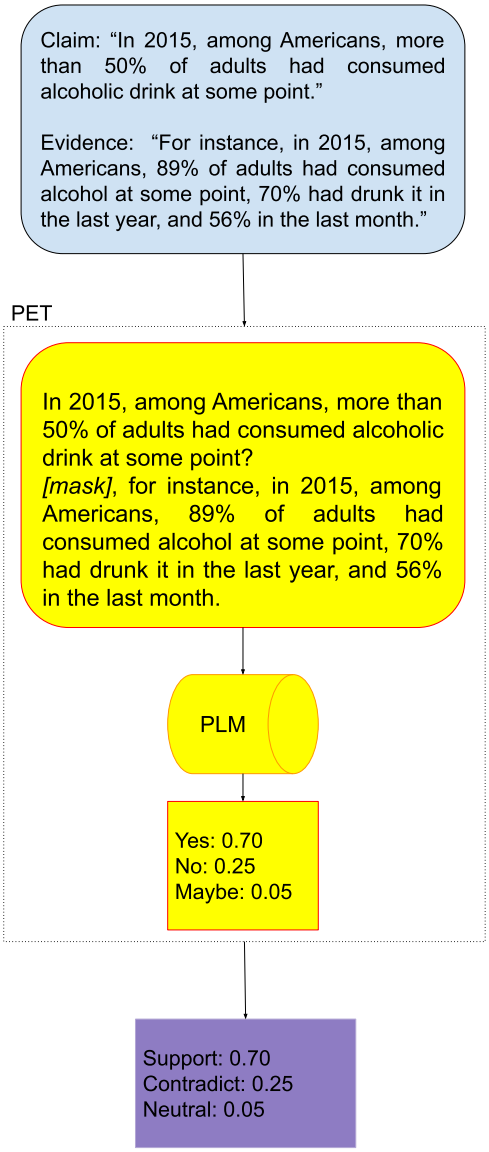

Figure 6.3: An example of doing claim verification with PET.

learning, but always train the initial PLMs which have never been trained on any fact-checking datasets.

6.3 Results

We next discuss the results of our experiments.

Results on SciFact_retrieved Figure 6.2 presents experimental results on SciFact, where the unlabelled pool is large, heavily imbalanced and the domain is technical. Each subfigure shows results for a different PET among the six under consideration.

Data retrieved with Active PETs brings substantial improvements for all of the models,

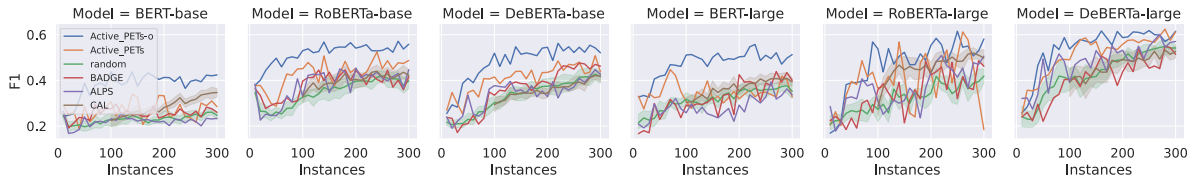

Figure 6.4: Few-Shot F1 Performance on cFEVER claim verification.

often from the very beginning but consistently as the number of shots increases from around 50 instances. Despite the performance fluctuations, training using data sampled with Active PETs rarely underperforms the baselines for SciFact. With Active PETs, Bert-base peaks at 0.352, RoBERTa-base peak at 0.345; DeBERTa-base peaks at 0.385; BERT-large peaks at 0.380; RoBERTa-large peaks at 0.409; DeBERTa-large peaks at 0.541. Generally, Active PETs shows a 10 to 20% increase in F1 scores, compared with various baselines.

Moreover, with Active PETs-o, i.e. when oversampling is further integrated with Active PETs, we observe a significant performance increase. Models tend to learn better from the beginning; the increase trend has less fluctuation; and the overall F1 scores are much higher. In this case, Bert-base peaks at 0.497, RoBERTa-base peak at 0.539; DeBERTa-base peaks at 0.551; BERT-large peaks at 0.548; RoBERTa-large peaks at 0.514; DeBERTa-large peaks at 0.587. This highlights the potential of oversampling, which increases the number of instances without additional labelling budget.

Among the baselines, we observe that training with data retrieved from all baselines failed to lead to any effective outcomes for BERT-base and DeBERTa-base within a labelling budget of 300 instances. While BADGE and CAL lead to some improvements over BERT-large and RoBERTa-large when given over 100 instances, random and ALPS failed to bring any improvements. Baseline results are better with RoBERTa-base and DeBERTa-large, but underperform Active PETs.

Results on cFEVER Figure 6.4 presents F1 scores on cFEVER, where the unlabelled pool is large, imbalanced and the domain is somewhat technical. In this case, models generally achieve higher F1 scores than on SciFact. First of all, we observe that Active PETs outperforms random baseline in a more stable manner. It is over 10% higher than random most of the time, although it shows large performance fluctuations on RoBERTa-large. With Active PETs, Bert-base peaks at 0.34, RoBERTa-base peak at 0.524; DeBERTa-base peaks at 0.508; BERT-large peaks at 0.447;

RoBERTa-large peaks at 0.612; DeBERTa-large peaks at 0.624. Moreover, Active PETs-o leads to a further performance boost, and more importantly, smooths out the large performance fluctuations. It is about 20% better than the random baseline most of the time. Specifically, Bert-base peaks at 0.438, RoBERTa-base peak at 0.571; DeBERTa-base peaks at 0.562; BERT-large peaks at 0.557; RoBERTa-large peaks at 0.615; DeBERTa-large peaks at 0.618.

When it comes to the baselines, the baselines do not struggle as much in the worst cases. Even if BERT-base's performance merely increased with most of the baselines, all of the other models managed to improve within the budget. With random sampling, RoBERTa-base, DeBERTa-base, BERT-large and RoBERTa-large all roughly peak at around 0.4, while DeBERTa-large is much better and peaks at around 0.5. BADGE, CAL and ALPS are in general better than random, but achieves lower F1 scores than Active PETs, especially in few-shot settings when the labelling budge is below 100.

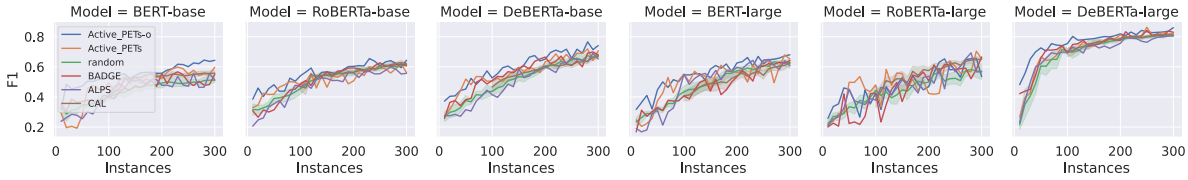

Figure 6.5: Few-Shot F1 Performance on SciFact_oracle claim verification.

6.4 Ablation Study

With SciFact we designed a slightly different pipeline where we conduct both evidence retrieval and claim verification – a setting that wasn't provided with cFEVER. To assess the impact of the addition of the evidence retrieval component on SciFact, we further perform ablation experiments on SciFact with oracle evidence, i.e., SciFact_oracle configuration.

With oracle evidence, the number of 'NOT_ENOUGH_INFO' claim-evidence pairs are significantly reduced, resulting in a more balanced overall label distribution. After reserving 100 instances from each class for the test set, the unlabelled pool has 765 instances in total, where 'SUPPORTS' takes 46.54%, 'NOT_ENOUGH_INFO' takes 38.43% and 'REFUTES' takes 15.03%. As shown in Figure 6.5, overall few-shot performance is much better and active learning demonstrates lesser performance gains. Sampling with baseline active learning strategies in general leads to similar results as random sampling. Surprisingly, coupling Active PETs

with oversampling when the labelled pool is reasonably balanced, still maintains performance advantages across models. Under this setting, Bert-base peaks at 0.645, RoBERTa-base peak at 0.655; DeBERTa-base peaks at 0.766; BERT-large peaks at 0.68; RoBERTa-large peaks at 0.657; DeBERTa-large peaks at 0.86.

As demonstrated above, active learning is much more helpful for SciFact in a real-world setting than in an oracle setting. We could expect that if this finding generalises to cFEVER, active learning in a real-world setting involving evidence retrieval could possibly lead to larger performance gains.

6.5 Analysis and Discussion

To better understand the impact of data prioritisation, we delve into the labelled data. In the interest of focus, we compare Active PETs with the SOTA cold-start method ALPS by analysing the best-performing PLM DeBERTa-large where 300 instances are selected.

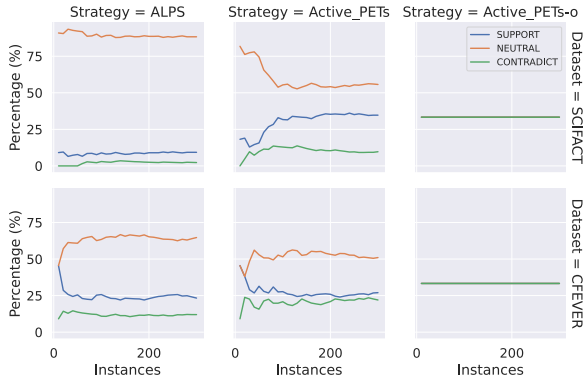

Figure 6.6: Label Distribution of data obtained with active learning by DeBERTa-large. The upper row is for SciFact_retrieved and the lower row is for cFEVER.

Balancing Effects We first look at the distribution of labels for the selected data. Figure 6.6 shows remarkable difference on label distribution for different active learning strategies. ALPS samples over 80% data from 'NOT_ENOUGH_INFO', less than 10% from 'SUPPORTS' and very few from 'REFUTES' for SciFact; over 60% data from 'NOT_ENOUGH_INFO', over 20% from 'SUPPORTS' and less than 20% from 'REFUTES' for cFEVER. They correlate well with original label distribution of each unlabelled pool, as presented in table 6.1. It suggests that ALPS is not sensitive to label distribution. However, Active PETs manages to sample a much more balanced distribution out of the extremely skewed original distribution. For SciFact, despite the initial few iterations, Active PETs samples less than 60% data from 'NOT_ENOUGH_INFO', less

than 40% data from 'SUPPORTS', around 10% data from 'REFUTES'; for cFEVER, Active PETs samples less than 60% data from 'NOT_ENOUGH_INFO', over 20% data from 'SUPPORTS', around 20% data from 'REFUTES'. In both datasets, label distribution from Active PETs are significantly more balanced than ALPS. Finally, the strategy with oversampling returns perfectly balanced distribution as expected. We identify a strong correlation between label distribution and classification performance.

Linguistic Effects Aiming at providing further insights into data quality, we conduct corpus-based linguistic analysis to investigate lexical richness and semantic similarity.

Lexical Richness			
	ALPS	**Active_PETs**	**Active_PETs-o**
SciFact_retrieved	0.0362	0.0387	0.0447
cFEVER	0.0389	0.0413	0.0503
Semantic Similarity			
	ALPS	**Active_PETs**	**Active_PETs-o**
SciFact_retrieved	0.7921	0.8031	0.8054
cFEVER	0.7449	0.7744	0.7841

Table 6.2: Lexical richness is measured with Maas Type-Token Ratio (MTTR) scores and Semantic Similarity is measured by cosine similarity scores on embeddings of claims and evidences.

Lexical Richness

A popular metric for calculating lexical richness is Type-Token Ratio (TTR), where the total number of unique tokens is divided by the total number of tokens. We use Maas Type-Token Ratio (Maas TTR) (Maas, 1972), a logarithmic variant of TTR, which is demonstrated to be less sensitive to the length of the text (McCarthy and Jarvis, 2007):

$$a^2 = \frac{logN - logV}{logN^2} \tag{6.2}$$

where N is the number of tokens in the corpus and V is the number of unique tokens in the corpus.

As shown in the upper part of Table 6.2, data selected by ALPS has the lowest lexical richness, while Active PETs leads to higher lexical richness for both datasets. Even more surprisingly, when integrating Active PETs with oversampling, the corpus has even higher score at lexical richness, despite that there are multiple duplicated instances in the corpus. One possibility is that training

data with higher lexical richness may convey more useful information, as a bigger vocabulary enables more precise expressions.

Semantic Similarity

To investigate the overall data diversity, we calculate the average semantic similarity of all possible claim-evidence pairs in the corpus.[3] We obtain embeddings of claims and evidences with the PLM at interest, namely DeBERTa-large that has been trained on MNLI. For each embedded claim, we calculate its cosine similarity score with all embedded evidences in the corpus. The average of all similarity scores is then obtained. The lower part of Table 6.2 shows that ALPS leads to lowest overall semantic embedding similarity scores and Active PETs leads to higher scores. Integrated with oversampling, Active PETs leads to even higher similarity scores. It correlates well with the design of the strategies: ALPS explicitly encourages data diversity, while Active PETs focuses on committee uncertainty. One possible explanation is that data diversity is not as beneficial when the unlabelled pool contains less relevant instances: in the case of SciFact and cFEVER datasets, the majority of the unlabelled pool belongs to the 'NOT_ENOUGH_INFO' class where the evidence is not enough to reach a verdict for the claim.

6.6 Summary

In this chapter, we have presented the first study on data annotation prioritisation for claim verification in automated fact-checking. With our novel method Active PETs, we demonstrate the potential of utilising a committee of PETs to collaboratively select unlabelled data for annotation, furthering in turn the extensibility of PET to active learning for the first time. Experiments on the SciFact and cFEVER datasets demonstrate the effectiveness of our proposed method, particularly in dealing with imbalanced data. Our proposed model consistently outperforms the random, BADGE, CAL and ALPS baselines by a margin. Further integration with an oversampling strategy that does not impact labelling effort leads to consistent performance improvements in all tested settings. Data that is more balanced shows to have higher lexical richness and semantic similarity, leading to better training results. While we have shown its effectiveness for claim verification here, in the future we aim to investigate Active PETs in other downstream tasks.

[3]Note that if we only calculate the retrieved pairs, the average similarity scores are approximately 1 for all strategies.

Chapter 7

Conclusions and Future Directions

Claim verification to support an automated fact-checking pipeline had been extensively studied when we embarked in this thesis, but there was a significant gap in developing and studying these models in the challenging settings posed by a few-shot learning scenario where very limited training data is available for training a model. Motivated by the need to develop claim verification components with the constraints of limited resources, and enabling the extensibility of automated fact-checking to new, emerging domains for which labeled data was scarce, this thesis investigated few-shot claim verification. The thesis aimed to study the extent of applicability of existing claim verification models, as well as to propose improved solutions to better tackle the problem.

Through this investigation, this study has made substantial contributions to the field of few-shot claim verification research, primarily in three key directions. First, we introduced SEED, a scalable few-shot claim verification method that requires no parameter update. Second, we proposed MAPLE, an efficient method that outperforms LLaMa 2 with only a T5-small model. And third, we proposed Active PETs, a novel ensemble active learning method facilitating few-shot data annotation prioritization. Below, we repeat the research questions and present our general findings to address them comprehensively.

7.1 General Findings

Referring back to the research questions we set forth in the introduction of the thesis, here we provide our answers based on what we have learned with the work conducted in this thesis.

RQ1: How do the challenges vary between different types of datasets, including domain-

specific versus more general, and synthetic versus non-synthetic data, as well as different dataset configurations such as oracle versus retrieved evidence configurations?

In examining the challenges associated with different datasets for few-shot claim verification, we found that general and synthetic datasets, such as FEVER sourced from the Wikipedia domain with claims mutated from source text, present fewer obstacles. Conversely, domain-specific datasets such as SciFact, focused on biomedical research, and Climate FEVER, dedicated to climate change claims, pose significantly greater challenges. Notably, methods exhibit optimal performance on the general Wikipedia domain, indicating the lower complexity of verifying more general claims. However, performance decreases notably when applied to more specialized domains, emphasizing the higher demands for specialized knowledge and contextual understanding in these areas.

Furthermore, the comparison between SciFact_oracle and Climate FEVER highlights the impact of data origin on verification difficulty. Despite both datasets featuring domain-specific claims, few-shot claim verification methods have higher performance on SciFact_oracle, comprised of manually mutated scientific claims, than Climate FEVER, which contains claims scraped from the web. This discrepancy underscores the challenges presented by real-world data, including noise and bias inherent in externally sourced claims. Thus, while synthetic data provides a controlled environment for verification, real-world data introduces complexity and challenges.

Additionally, the notably poorer performance of few-shot claim verification methods on the SciFact_retrieved configuration underscores the difficulty of verifying claims with retrieved noisy evidence. This finding highlights the considerable challenges posed by noisy evidence in real-world scenarios for automated verification methods.

Overall, the findings underscore the significant challenges posed by specific domains, noisy claims, and imperfect evidence in few-shot claim verification scenarios.

RQ2: What are the existing and novel few-shot claim verification methods, and how do they tackle the obstacles presented by scarce annotations, tight annotation budgets, and the limitations imposed by restricted computing resources?

Throughout this thesis, various few-shot claim verification methods have been introduced to tackle the challenges posed by limited annotations, tight annotation budgets, and constrained computing resources. While the Perplexity-based method stands as the sole existing approach in the few-shot claim verification literature, PET and LLaMA 2 are established few-shot methods

in the broader NLP community, adapted here for claim verification purposes. Importantly, we propose three novel methods for claim verification: SEED, MAPLE, and Active PETs.

To overcome the hurdles presented by sparse annotations, stringent annotation budgets, and limited computing resources, the Perplexity-based method utilizes perplexity scores and manual thresholds. SEED generates class representative vectors, PET crafts natural language patterns and verbalizers, while LLaMA 2 enable classifications through prompting. MAPLE leverages in-domain seq2seq training and explores its convergence process, whereas Active PETs integrate ensemble-based active learning with PET for overall optimization.

RQ3: What are the comparative strengths and weaknesses of various few-shot claim verification methods, and which method is most suitable for specific scenarios?

Here we elaborate on the strengths and weaknesses of the aforementioned few-shot claim verification methods to offer tailored recommendations for various scenarios.

Despite its relatively high interpretability and low computing cost, the Perplexity-based approach is generally not recommended due to its limitation to binary classification and being surpassed by various other methods.

Prompting LLaMa 2 requires minimal setup efforts, as many platforms host such models for customer chatting purposes. However, deploying this method on a large scale would encounter several challenges: the "no response" problem, high costs, significant demands on computational resources, limited availability of labeled data, and providing only average performance. It is advisable for quick prototyping of one-shot scenarios with low expectations for scaling and long-term use.

SEED stands out for its simplicity: it is conceptually easy to understand, requires minimal setup, has low computational demands, and scales particularly well for inference tasks. However, it relies on NLI-trained Pre-trained Language Models (PLMs) and may not achieve the best performance, especially in one-shot cases. It is recommended for settings with 5-20 shots, where an NLI-trained PLM is available, and the target domain data is relatively similar to a standard NLI dataset.

Generally, MAPLE emerges as the best-performing method within five shots and demonstrates robustness across different dataset domains and configurations. Unlike PET or SEED, it does not rely on NLI-trained PLMs. However, MAPLE relies on unlabeled data for seq2seq in-domain training. Although it excels within five shots, its performance does not improve beyond

this threshold. It is recommended when unlabeled data and some computational resources are available, and labeled data is extremely limited.

Alternatively, PET is the only method directly suitable for higher-shot scenarios, as it can learn from hundreds of annotated data points. However, when thousands of labeled data points are available, PET may not necessarily outperform supervised fine-tuning methods.

The flexibility of PET in learning from higher-shot data is particularly desirable, especially when the annotation budget is less limiting. In such cases, prioritizing data annotation to optimize the use of the budget can greatly benefit overall performance. Active PETs effectively merge ensemble-based active learning with PET to address few-shot claim verification challenges and achieve significant overall improvements. This approach is particularly effective when the unlabeled pool exhibits heavily skewed data distribution for three-way claim verification, a common scenario in real-world fact-checking.

7.2 Limitations and Future Directions

While this thesis presents a collection of innovative contributions to automated fact-checking, potentials remain to be explored. Here we discuss limitations.

While our research encompasses datasets from multiple domains, it is limited to experiments conducted on English texts. Recently published datasets have introduced content in other languages, such as Chinese (Hu et al., 2022b), Danish (Nørregaard and Derczynski, 2021) and Farsi (Zarharan et al., 2021) and many more (Gupta and Srikumar, 2021), and have expanded to include multi-modal options, such as tabular (Aly et al., 2021), image (Gupta et al., 2022b) and video data (Liu et al., 2023). Consequently, the generalizability of our methods to multi-lingual and multi-modal contexts is yet to be explored and validated.

SEED and MAPLE, specialized for few-shot claim verification, exhibit quick convergence, making them suitable for specific scenarios. However, further research is needed to extend their applicability to higher-shot settings. Active PETs, successfully proposing an ensemble method on pretrained language models of similar size, face challenges to include models of varying size in the same committee with its current voting mechanism.

While the focus of this thesis is on proposing models based on BERT-sized models for accessibility and scalability, future work could explore the temporary inclusion of LLMs to harness their capabilities and apply techniques such as knowledge distillation and model quantization, to

reduce the model size for long-term deployment. Apart from these specific extensions mentioned, we also propose a few board directions to inspire future work below.

7.2.1 Unified Benchmark for Automated Fact-Checking

The evolving landscape of misinformation necessitates a unified benchmark for automated fact-checking that is both comprehensive and adaptable to the diverse nature of claims encountered across different spheres of information. A benchmark with a multi-domain, multi-lingual, and multi-modal approach addresses this need by accommodating the wide spectrum of misinformation that proliferates across various subject matters, languages, and formats. Such a benchmark would not only support the evaluation of fact-checking systems in a more global context but also ensure that these systems are robust and versatile enough to handle the complexity of information as it exists in the real world. By covering a broad range of data types and sources, this benchmark would push the boundaries of current fact-checking methodologies, encouraging advancements that are capable of tackling the nuanced and varied nature of false information.

Building on the foundation of a multi-faceted benchmark, it is crucial to enhance the integration of claim detection and claim validation processes, focusing on the concept of claim checkworthiness as a measure of societal impact. This approach proposes a tighter integrity between the two fundamental components of automated fact-checking, ensuring that the systems not only identify potentially misleading claims but also prioritize them based on their significance and potential impact on society. By utilizing the checkworthiness of claims to evaluate the effectiveness of a claim validation system, this integrated benchmark aims to direct fact-checking resources more efficiently, focusing efforts on claims that, if left unverified, could have detrimental effects on public understanding and discourse. Such an integrated approach underscores the importance of not just detecting and validating claims in isolation but doing so in a manner that reflects the real-world implications and priorities of fact-checking in the digital age.

7.2.2 Claim Verification-Centered System Design for Automated Fact-Checking

The task of evidence retrieval can be fulfilled with a rough retriever and a classifier, as demonstrated by the QMUL@SciVer study in Appendix A. Given that a claim verification model is essentially a NLI model that does three-way classification, it is plausible to design a claim validation system centered on a scalable claim verification model. For example, train a NLI model with carefully designed sampling strategy on a claim validation dataset and use it to directly run inference on

every candidate evidence in the corpus to get the veracity label. Such simplified system design has better integrity and does not experience accumulated errors in a pipeline system. Another potential direction is to use the training data inversely: first training a claim verification model with oracle evidence; then use it as a reward model to train an evidence retrieval model with reinforcement learning such that the retrieved evidence module is optimized for the downstream claim verification module. We believe a more robust system design with better integrity would be greatly beneficial to the overall performance.

7.2.3 Human and Model Collaborative Workflow

Active PETs demonstrates using models to help with data annotation prioritization in experimental settings. It would be great to deploy such active learning functionality in real-world fact-checking practices. While active learning focuses on selecting the best data out of the unlabeled pool, the best data is not necessarily in the pool but may be generatable given the pool. An interesting direction is to extend the scope of active learning from a selecting/ranking problem to a generative problem. For example, after top-k samples are selected, we can incorporate an additional step to cluster and summarise the top-k samples into just fewer samples to reduce the annotation workload. Another possibility is to use data augmentation techniques to populate out the top-k selected samples, which can then be used as an extension of the selected samples to do more fine-grained data selection or as additional training data for the current iteration with labels inherited from the top-k samples. The extended workflow could include selecting, summarising, annotating, populating, and training, using various NLP methods.

Alternatively, when training a capable system to conduct automated fact-checking is out of the scope due to limited time, resources, and applications, a system that co-inference with input from a human worker would also be helpful. For example, if a fact-checking organization tends to have five fact-checkers examining the same claim, we may reduce the number to three when an automated fact-checking system confirms the human judgments in the early stages. Otherwise, deploying a double-checking system on human judgments would also provide valuable help. A collaborative framework between human workers and models would have great contributions to leveraging the strengths of both for enhanced efficiency and trust.

7.2.4 Beyond Automated Fact-Checking

Adapting MAPLE into an Evaluation Metric While proposing MAPLE, we introduced SemSim as an NLG evaluation metric. Meanwhile, considering its unique capabilities in few-shot claim verification, MAPLE itself also has great potential as a general NLG evaluation metric, either unsupervised or with few-shot supervision. The exploration of adapting MAPLE into an evaluation metric for broader applications is promising.

Applying Claim Verification to LLM Performance Evaluation Though automated fact-checking was proposed to primarily address online misinformation problems, it is directly transferable to be applied to model hallucination detection for Gen AI. Current mainstream Gen AI inference framework uses RAG, which first retrieves relevant information from a given corpus, and then generates responses based on the revised prompt. We can simply treat the retrieved information by RAG as the evidence and the generated response as the claim, and run a claim verification model on the pair. Given the growing public concerns about AI safety, it would be beneficial to the NLP community to include claim verification as part of standard LLM performance evaluation.

7.3 Summary

In this chapter, we have addressed the research questions by presenting our findings and have outlined limitations and future directions for few-shot claim verification research.

We have emphasized the challenges posed by specific domains, noisy claims, and imperfect evidence in real-world scenarios. Our analysis has revealed that general and synthetic datasets, such as FEVER, pose fewer challenges compared to domain-specific and more natural datasets such as SciFact and Climate FEVER. Additionally, verifying claims with retrieved noisy evidence, especially in configurations like SciFact_retrieved, has proven to be significantly more challenging for automated methods.

Furthermore, we have provided an overview of the introduced few-shot claim verification methods, including the Perplexity-based method, PET, LLaMA 2, SEED, MAPLE, and Active PETs. Insights into the strengths and weaknesses of these methods have been provided. SEED is recommended for 5-20 shots, MAPLE has demonstrated robustness within five shots, and Active PETs are suitable for combining data annotation prioritization. Various other baseline methods also have their optimal use cases.

Looking forward, we also outlined potential future research directions, including the construc-

tion of unified benchmark for automated fact-checking, the development of claim verification-centered system designs for automated fact-checking and the exploration of human-model collaborative workflows. Furthermore, we discussed the potential adaptation of MAPLE into a general NLG evaluation metric and applying claim verification to LLMs evaluations.

In summary, our findings offer valuable insights into the intricate landscape of few-shot claim verification research, shedding light on both the challenges and opportunities inherent in this domain. By meticulously exploring the nuances of various datasets and dataset configurations, as well as scrutinizing the performance of established and novel few-shot claim verification methods, we have uncovered crucial factors influencing the effectiveness of automated fact-checking systems. These insights not only deepen our understanding of the complexities involved but also lay the groundwork for future advancements in automated fact-checking and natural language processing. Through continued exploration and innovation, we can further refine and optimize few-shot claim verification methodologies, ultimately enhancing the automated fact-checking systems in combating misinformation in the digital age.

Bibliography

Ahmed Abdelali, Kareem Darwish, Nadir Durrani, and Hamdy Mubarak. 2016. Farasa: A fast and furious segmenter for Arabic. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, pages 11–16, San Diego, California. Association for Computational Linguistics.

Heike Adel. 2018. *Deep learning methods for knowledge base population*. Text.PhDThesis, Ludwig-Maximilians-Universität München. ISSN: 1922-8945.

Firoj Alam, Alberto Barrón-Cedeño, Gullal S Cheema, Sherzod Hakimov, Maram Hasanain, Chengkai Li, Rubén Míguez, Hamdy Mubarak, Gautam Kishore Shahi, Wajdi Zaghouani, et al. 2023. Overview of the clef-2023 checkthat! lab task 1 on check-worthiness in multimodal and multigenre content. *Working Notes of CLEF*.

Tariq Alhindi, Savvas Petridis, and Smaranda Muresan. 2018. Where is your evidence: Improving fact-checking by justification modeling. In *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*, pages 85–90, Brussels, Belgium. Association for Computational Linguistics.

Liesbeth Allein and Marie-Francine Moens. 2020. Checkworthiness in automatic claim detection models: Definitions and analysis of datasets. In *Multidisciplinary International Symposium on Disinformation in Open Online Media*, pages 1–17. Springer.

Rami Aly, Zhijiang Guo, Michael Sejr Schlichtkrull, James Thorne, Andreas Vlachos, Christos Christodoulopoulos, Oana Cocarascu, and Arpit Mittal. 2021. The fact extraction and VERification over unstructured and structured information (FEVEROUS) shared task. In *Proceedings of the Fourth Workshop on Fact Extraction and VERification (FEVER)*, pages 1–13, Dominican Republic. Association for Computational Linguistics.

Fatma Arslan, Naeemul Hassan, Chengkai Li, and Mark Tremayne. 2020. A benchmark dataset of check-worthy factual claims. *Proceedings of the International AAAI Conference on Web and Social Media*, 14(1):821–829.

Jordan T. Ash, Chicheng Zhang, Akshay Krishnamurthy, John Langford, and Alekh Agarwal. 2020. Deep batch active learning by diverse, uncertain gradient lower bounds. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Pepa Atanasova, Alberto Barron-Cedeno, Tamer Elsayed, Reem Suwaileh, Wajdi Zaghouani, Spas Kyuchukov, Giovanni Da San Martino, and Preslav Nakov. 2018. Overview of the CLEF-2018 CheckThat! Lab on Automatic Identification and Verification of Political Claims. Task 1: Check-Worthiness. *arXiv:1808.05542 [cs]*. ArXiv: 1808.05542.

Pepa Atanasova, Preslav Nakov, Georgi Karadzhov, Mitra Mohtarami, and Giovanni Da San Martino. 2019a. Overview of the clef-2019 checkthat! lab: Automatic identification and verification of claims. task 1: Check-worthiness. In *CLEF (Working Notes)*.

Pepa Atanasova, Preslav Nakov, Lluís Màrquez, Alberto Barrón-Cedeño, Georgi Karadzhov, Tsvetomila Mihaylova, Mitra Mohtarami, and James Glass. 2019b. Automatic fact-checking using context and discourse information. *J. Data and Information Quality*, 11(3).

Pepa Atanasova, Jakob Grue Simonsen, Christina Lioma, and Isabelle Augenstein. 2020. Generating fact checking explanations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7352–7364, Online. Association for Computational Linguistics.

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. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4685–4697, Hong Kong, China. Association for Computational Linguistics.

Michael Azmy, Peng Shi, Jimmy Lin, and Ihab Ilyas. 2018. Farewell Freebase: Migrating the SimpleQuestions dataset to DBpedia. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2093–2103, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Mevan Babakar and Will Moy. 2016. The State of Automated Factchecking. Technical report, Full Fact, London, UK.

Krisztian Balog. 2018. Populating Knowledge Bases. In Krisztian Balog, editor, *Entity-Oriented Search*, The Information Retrieval Series, pages 189–222. Springer International Publishing, Cham.

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan. Association for Computational Linguistics.

Roy Bar-Haim, Ido Dagan, Bill Dolan, Lisa Ferro, and Danilo Giampiccolo. 2006. The second PASCAL recognising textual entailment challenge. *Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment*.

Roy Bar-Haim, Ido Dagan, and Idan Szpektor. 2014. Benchmarking Applied Semantic Inference: The PASCAL Recognising Textual Entailment Challenges. In Nachum Dershowitz and Ephraim Nissan, editors, *Language, Culture, Computation. Computing - Theory and Technology: Essays Dedicated to Yaacov Choueka on the Occasion of His 75th Birthday, Part I*, Lecture Notes in Computer Science, pages 409–424. Springer, Berlin, Heidelberg.

Alberto Barrón-Cedeño, Tamer Elsayed, Preslav Nakov, Giovanni Da San Martino, Maram Hasanain, Reem Suwaileh, Fatima Haouari, Nikolay Babulkov, Bayan Hamdan, Alex Nikolov, Shaden Shaar, and Zien Sheikh Ali. 2020. Overview of checkthat! 2020: Automatic identification and verification of claims in social media. In *Experimental IR Meets Multilinguality, Multimodality, and Interaction: 11th International Conference of the CLEF Association, CLEF 2020, Thessaloniki, Greece, September 22–25, 2020, Proceedings*, page 215–236, Berlin, Heidelberg. Springer-Verlag.

Alberto Barrón-Cedeño, Firoj Alam, Andrea Galassi, Giovanni Da San Martino, Preslav Nakov, , Tamer Elsayed, Dilshod Azizov, Tommaso Caselli, Gullal S. Cheema, Fatima Haouari, Maram Hasanain, Mucahid Kutlu, Chengkai Li, Federico Ruggeri, Julia Maria Struß, and Wajdi Zaghouani. 2023. Overview of the CLEF–2023 CheckThat! Lab checkworthiness, subjectivity, political bias, factuality, and authority of news articles and their source. In

Experimental IR Meets Multilinguality, Multimodality, and Interaction. Proceedings of the Fourteenth International Conference of the CLEF Association (CLEF 2023).

Bettina Berendt, Peter Burger, Rafael Hautekiet, Jan Jagers, Alexander Pleijter, and Peter Van Aelst. 2021. Factrank: Developing automated claim detection for dutch-language fact-checkers. *Online Social Networks and Media*, 22:100113.

Johan Bos and Katja Markert. 2006. When logical inference helps determining textual entailment (and when it doesn't). *Proceedings of the second PASCAL RTE challenge*, page 26.

Mostafa Bouziane, Hugo Perrin, Aurelien Cluzeau, Julien Mardas, and Amine Sadeq. 2020. Team Buster.ai at CheckThat! 2020: Insights And Recommendations To Improve Fact-Checking. In *CLEF (Working Notes)*, page 12.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Abhijnan Chakraborty, Bhargavi Paranjape, Sourya Kakarla, and Niloy Ganguly. 2016. Stop clickbait: Detecting and preventing clickbaits in online news media. In *2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pages 9–16.

Nathanael Chambers, Daniel Cer, Trond Grenager, David Hall, Chloe Kiddon, Bill MacCartney, Marie-Catherine de Marneffe, Daniel Ramage, Eric Yeh, and Christopher D. Manning. 2007. Learning alignments and leveraging natural logic. In *Proceedings of the ACL-PASCAL*

Workshop on Textual Entailment and Paraphrasing, pages 165–170, Prague. Association for Computational Linguistics.

Lingjiao Chen, Matei Zaharia, and James Zou. 2024. How is chatgpt's behavior changing over time? *Harvard Data Science Review*. Https://hdsr.mitpress.mit.edu/pub/y95zitmz.

Qingyu Chen, Yifan Peng, and Zhiyong Lu. 2019a. BioSentVec: creating sentence embeddings for biomedical texts. *2019 IEEE International Conference on Healthcare Informatics (ICHI)*, pages 1–5. ArXiv: 1810.09302.

Sihao Chen, Daniel Khashabi, Wenpeng Yin, Chris Callison-Burch, and Dan Roth. 2019b. 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.

Yimin Chen, Niall J. Conroy, and Victoria L. Rubin. 2015. Misleading Online Content: Recognizing Clickbait as "False News". In *Proceedings of the 2015 ACM on Workshop on Multimodal Deception Detection*, WMDD '15, pages 15–19, New York, NY, USA. Association for Computing Machinery.

Gennaro Chierchia and Sally McConnell-Ginet. 2000. *Meaning and Grammar: An Introduction to Semantics*. MIT Press. Google-Books-ID: pxJGet3pKdoC.

Alina Maria Ciobanu and Liviu P. Dinu. 2018. Simulating language evolution: a tool for historical linguistics. In *Proceedings of the 27th International Conference on Computational Linguistics: System Demonstrations*, pages 68–72, Santa Fe, New Mexico. Association for Computational Linguistics.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: pre-training text encoders as discriminators rather than generators. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Sarah Cohen, Chengkai Li, Jun Yang, and Cong Yu. 2011. Computational journalism: A call to arms to database researchers. In *CIDR 2011, Fifth Biennial Conference on Innovative Data*

Systems Research, Asilomar, CA, USA, January 9-12, 2011, Online Proceedings, pages 148–151. www.cidrdb.org.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. 2017. Supervised learning of universal sentence representations from natural language inference data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 670–680, Copenhagen, Denmark. Association for Computational Linguistics.

Ido Dagan, Bill Dolan, Bernardo Magnini, and Dan Roth. 2009. Recognizing textual entailment: Rational, evaluation and approaches. *Natural Language Engineering*, 15(4):i–xvii. Publisher: Cambridge University Press.

Ido Dagan and Sean P. Engelson. 1995. Committee-Based Sampling For Training Probabilistic Classifiers. In *In Proceedings of the Twelfth International Conference on Machine Learning*, pages 150–157. Morgan Kaufmann.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2005. The PASCAL Recognising Textual Entailment Challenge. In *Machine Learning Challenges. Evaluating Predictive Uncertainty, Visual Object Classification, and Recognising Tectual Entailment*, Lecture Notes in Computer Science, pages 177–190, Berlin, Heidelberg. Springer.

Marie-Catherine de Marneffe, Anna N. Rafferty, and Christopher D. Manning. 2008. Finding contradictions in text. In *Proceedings of ACL-08: HLT*, pages 1039–1047, Columbus, Ohio. Association for Computational Linguistics.

Rodrigo De Salvo Braz, Roxana Girju, Vasin Punyakanok, Dan Roth, and Mark Sammons. 2005. An inference model for semantic entailment in natural language. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence*, IJCAI'05, page 1678–1679, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Leon Derczynski, Kalina Bontcheva, Maria Liakata, Rob Procter, Geraldine Wong Sak Hoi, and Arkaitz Zubiaga. 2017. SemEval-2017 task 8: RumourEval: Determining rumour veracity and support for rumours. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 69–76, Vancouver, Canada. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

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.

Nan Duan, Duyu Tang, and Ming Zhou. 2020. Machine reasoning: Technology, dilemma and future. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Tutorial Abstracts*, pages 1–6, Online. Association for Computational Linguistics.

Liat Ein-Dor, Alon Halfon, Ariel Gera, Eyal Shnarch, Lena Dankin, Leshem Choshen, Marina Danilevsky, Ranit Aharonov, Yoav Katz, and Noam Slonim. 2020. Active Learning for BERT: An Empirical Study. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7949–7962, Online. Association for Computational Linguistics.

Henry Elder, Jennifer Foster, James Barry, and Alexander O'Connor. 2019. Designing a symbolic intermediate representation for neural surface realization. In *Proceedings of the Workshop on Methods for Optimizing and Evaluating Neural Language Generation*, pages 65–73, Minneapolis, Minnesota. Association for Computational Linguistics.

Tamer Elsayed, Preslav Nakov, Alberto Barrón-Cedeño, Maram Hasanain, Reem Suwaileh, Giovanni Da San Martino, and Pepa Atanasova. 2019. Overview of the clef-2019 checkthat! lab: Automatic identification and verification of claims. In *Experimental IR Meets Multilinguality, Multimodality, and Interaction: 10th International Conference of the CLEF Association, CLEF 2019, Lugano, Switzerland, September 9–12, 2019, Proceedings*, page 301–321, Berlin, Heidelberg. Springer-Verlag.

Yufei Feng, Zi'ou Zheng, Quan Liu, Michael Greenspan, and Xiaodan Zhu. 2020. Exploring end-to-end differentiable natural logic modeling. In *Proceedings of the 28th International*

Conference on Computational Linguistics, pages 1172–1185, Barcelona, Spain (Online). International Committee on Computational Linguistics.

William Ferreira and Andreas Vlachos. 2016. Emergent: a novel data-set for stance classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1163–1168, San Diego, California. Association for Computational Linguistics.

Diane Francis and Full Fact. 2016. Fast & furious fact check challenge. https://www.herox.com/factcheck/teams.

Yoav Freund and David Haussler. 1997. Selective sampling using the query by committee algorithm. In *Machine Learning*, pages 133–168.

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. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 6407–6414. AAAI Press.

Artur d'Avila Garcez, Sebastian Bader, Howard Bowman, Luis C Lamb, Leo de Penning, BV Illuminoo, Hoifung Poon, and COPPE Gerson Zaverucha. 2022. Neural-symbolic learning and reasoning: A survey and interpretation. *Neuro-Symbolic Artificial Intelligence: The State of the Art*, 342(1):327.

Pepa Gencheva, Preslav Nakov, Lluís Màrquez, Alberto Barrón-Cedeño, and Ivan Koychev. 2017. A context-aware approach for detecting worth-checking claims in political debates. In *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017*, pages 267–276, Varna, Bulgaria. INCOMA Ltd.

Danilo Giampiccolo, H. Dang, B. Magnini, I. Dagan, Elena Cabrio, and W. Dolan. 2008. The Fourth PASCAL Recognizing Textual Entailment Challenge. In *TAC*.

Danilo Giampiccolo, Bernardo Magnini, Ido Dagan, and Bill Dolan. 2007. The third PASCAL recognizing textual entailment challenge. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pages 1–9, Prague. Association for Computational Linguistics.

Genevieve Gorrell, Kalina Bontcheva, Leon Derczynski, Elena Kochkina, Maria Liakata, and Arkaitz Zubiaga. 2018. RumourEval 2019: Determining Rumour Veracity and Support for Rumours. *arXiv:1809.06683 [cs]*. ArXiv: 1809.06683.

Zhijiang Guo, Michael Schlichtkrull, and Andreas Vlachos. 2022. A survey on automated fact-checking. *Transactions of the Association for Computational Linguistics*, 10:178–206.

Ashim Gupta and Vivek Srikumar. 2021. X-fact: A new benchmark dataset for multilingual fact checking. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 675–682, Online. Association for Computational Linguistics.

Prakhar Gupta, Chien-Sheng Wu, Wenhao Liu, and Caiming Xiong. 2022a. DialFact: A benchmark for fact-checking in dialogue. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3785–3801, Dublin, Ireland. Association for Computational Linguistics.

Vipin Gupta, Rina Kumari, Nischal Ashok, Tirthankar Ghosal, and Asif Ekbal. 2022b. MMM: An emotion and novelty-aware approach for multilingual multimodal misinformation detection. In *Findings of the Association for Computational Linguistics: AACL-IJCNLP 2022*, pages 464–477, Online only. Association for Computational Linguistics.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, Online. Association for Computational Linguistics.

Andreas Hanselowski. 2020. *A Machine-Learning-Based Pipeline Approach to Automated Fact-Checking*. Ph.D. Thesis, Technische Universität, Darmstadt.

Andreas Hanselowski, Christian Stab, Claudia Schulz, Zile Li, and Iryna Gurevych. 2019. A richly annotated corpus for different tasks in automated fact-checking. In *Proceedings of the*

23rd Conference on Computational Natural Language Learning (CoNLL), pages 493–503, Hong Kong, China. Association for Computational Linguistics.

Maram Hasanain and Tamer Elsayed. 2020. bigIR at CheckThat! 2020: Multilingual BERT for Ranking Arabic Tweets by Check-worthiness. In *CLEF (Working Notes)*.

Maram Hasanain, Fatima Haouari, Reem Suwaileh, Zien Sheikh Ali, Bayan Hamdan, Tamer Elsayed, A. Barrón-Cedeño, Giovanni Da San Martino, and Preslav Nakov. 2020. Overview of checkthat! 2020i arabic: Automatic identification and verification of claims in social media. In *CLEF*.

Naeemul Hassan, Fatma Arslan, Chengkai Li, and Mark Tremayne. 2017a. Toward automated fact-checking: Detecting check-worthy factual claims by claimbuster. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Halifax, NS, Canada, August 13 - 17, 2017*, pages 1803–1812. ACM.

Naeemul Hassan, Chengkai Li, and Mark Tremayne. 2015. Detecting check-worthy factual claims in presidential debates. In *Proceedings of the 24th ACM International Conference on Information and Knowledge Management, CIKM 2015, Melbourne, VIC, Australia, October 19 - 23, 2015*, pages 1835–1838. ACM.

Naeemul Hassan, Gensheng Zhang, Fatma Arslan, Josue Caraballo, Damian Jimenez, Siddhant Gawsane, Shohedul Hasan, Minumol Joseph, Aaditya Kulkarni, Anil Kumar Nayak, Vikas Sable, Chengkai Li, and Mark Tremayne. 2017b. ClaimBuster: the first-ever end-to-end fact-checking system. *Proceedings of the VLDB Endowment*, 10(12):1945–1948.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. DeBERTa: Decoding-enhanced BERT with Disentangled Attention. *arXiv:2006.03654 [cs]*. ArXiv: 2006.03654.

Christopher Hidey and Mona Diab. 2018. Team SWEEPer: Joint sentence extraction and fact checking with pointer networks. In *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*, pages 150–155, Brussels, Belgium. Association for Computational Linguistics.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022a. Lora: Low-rank adaptation of large language models. In *ICLR 2022*.

Xuming Hu, Zhijiang Guo, GuanYu Wu, Aiwei Liu, Lijie Wen, and Philip Yu. 2022b. CHEF: A pilot Chinese dataset for evidence-based fact-checking. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3362–3376, Seattle, United States. Association for Computational Linguistics.

Zhiting Hu, Xuezhe Ma, Zhengzhong Liu, Eduard Hovy, and Eric Xing. 2016. Harnessing deep neural networks with logic rules. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2410–2420, Berlin, Germany. Association for Computational Linguistics.

Nathalie Japkowicz. 2000. The Class Imbalance Problem: Significance and Strategies. In *In Proceedings of the 2000 International Conference on Artificial Intelligence (ICAI*, pages 111–117.

Israa Jaradat, Pepa Gencheva, Alberto Barrón-Cedeño, Lluís Màrquez, and Preslav Nakov. 2018. ClaimRank: Detecting check-worthy claims in Arabic and English. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, pages 26–30, New Orleans, Louisiana. Association for Computational Linguistics.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling Laws for Neural Language Models. *arXiv:2001.08361 [cs, stat]*. ArXiv: 2001.08361.

Rhea Kapur and Phillip Rogers. 2020. Modeling language evolution and feature dynamics in a realistic geographic environment. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 788–798, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Yavuz Selim Kartal and Mucahid Kutlu. 2020. TrClaim-19: The first collection for Turkish check-worthy claim detection with annotator rationales. In *Proceedings of the 24th Conference on Computational Natural Language Learning*, pages 386–395, Online. Association for Computational Linguistics.

Lev Konstantinovskiy, Oliver Price, Mevan Babakar, and Arkaitz Zubiaga. 2021. Toward automated factchecking: Developing an annotation schema and benchmark for consistent automated claim detection. *Digital Threats*, 2(2).

Neema Kotonya and Francesca Toni. 2020. 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.

Amrith Krishna, Sebastian Riedel, and Andreas Vlachos. 2022. ProoFVer: Natural logic theorem proving for fact verification. *Transactions of the Association for Computational Linguistics*, 10:1013–1030.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2020. BioBERT: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240. Publisher: Oxford Academic.

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.

Torvald Lekvam, Björn Gambäck, and Lars Bungum. 2014. Agent-based modeling of language evolution. In *Proceedings of the 5th Workshop on Cognitive Aspects of Computational Language Learning (CogACLL)*, pages 49–54, Gothenburg, Sweden. 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.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Fuxiao Liu, Yaser Yacoob, and Abhinav Shrivastava. 2023. COVID-VTS: Fact extraction and verification on short video platforms. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 178–188, Dubrovnik, Croatia. Association for Computational Linguistics.

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.

H.D. Maas. 1972. Uber den Zusammenhang zwischen Wortschatzumfang und Lange eines Textes. *Springer*, 8:73–96.

Bill MacCartney and Christopher D. Manning. 2009. An extended model of natural logic. In *Proceedings of the Eight International Conference on Computational Semantics*, pages 140–156, Tilburg, The Netherlands. Association for Computational Linguistics.

Christopher Malon. 2018. Team papelo: Transformer networks at FEVER. In *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*, pages 109–113, Brussels, Belgium. Association for Computational Linguistics.

Katerina Margatina, Giorgos Vernikos, Loïc Barrault, and Nikolaos Aletras. 2021. Active Learning by Acquiring Contrastive Examples. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 650–663, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Philip M. McCarthy and Scott Jarvis. 2007. vocd: A theoretical and empirical evaluation. *Language Testing*, 24(4):459–488. Publisher: SAGE Publications Ltd.

Pablo Mendes, Max Jakob, and Christian Bizer. 2012. DBpedia: A multilingual cross-domain knowledge base. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 1813–1817, Istanbul, Turkey. European Language Resources Association (ELRA).

Filipe Mesquita, Matteo Cannaviccio, Jordan Schmidek, Paramita Mirza, and Denilson Barbosa. 2019. KnowledgeNet: A benchmark dataset for knowledge base population. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 749–758, Hong Kong, China. Association for Computational Linguistics.

Ndapandula Nakashole and Gerhard Weikum. 2012. Real-time population of knowledge bases: Opportunities and challenges. In *Proceedings of the Joint Workshop on Automatic Knowl-*

edge Base Construction and Web-scale Knowledge Extraction (AKBC-WEKEX), pages 41–45, Montréal, Canada. Association for Computational Linguistics.

Preslav Nakov, Alberto Barrón-Cedeño, Giovanni Da San Martino, Firoj Alam, Julia Maria Struß, Thomas Mandl, Rubén Míguez, Tommaso Caselli, Mucahid Kutlu, Wajdi Zaghouani, Chengkai Li, Shaden Shaar, Gautam Kishore Shahi, Hamdy Mubarak, Alex Nikolov, Nikolay Babulkov, Yavuz Selim Kartal, Javier Beltrán, Michael Wiegand, Melanie Siegel, and Juliane Köhler. 2022. Overview of the CLEF-2022 CheckThat! lab on fighting the COVID-19 infodemic and fake news detection. In *Proceedings of the 13th International Conference of the CLEF Association: Information Access Evaluation meets Multilinguality, Multimodality, and Visualization*, CLEF~'2022, Bologna, Italy.

Preslav Nakov, Alberto Barrón-Cedeño, Tamer Elsayed, Reem Suwaileh, Lluís Màrquez, Wajdi Zaghouani, Pepa Atanasova, Spas Kyuchukov, and Giovanni Da San Martino. 2018. Overview of the CLEF-2018 CheckThat! Lab on Automatic Identification and Verification of Political Claims. In *Experimental IR Meets Multilinguality, Multimodality, and Interaction*, pages 372–387, Cham. Springer International Publishing.

Preslav Nakov, David Corney, Maram Hasanain, Firoj Alam, Tamer Elsayed, Alberto Barrón-Cedeño, Paolo Papotti, Shaden Shaar, and Giovanni Da San Martino. 2021a. Automated fact-checking for assisting human fact-checkers. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 4551–4558. International Joint Conferences on Artificial Intelligence Organization. Survey Track.

Preslav Nakov, Giovanni Da San Martino, Tamer Elsayed, Alberto Barrón-Cedeño, Rubén Míguez, Shaden Shaar, Firoj Alam, Fatima Haouari, Maram Hasanain, Nikolay Babulkov, Alex Nikolov, Gautam Kishore Shahi, Julia Maria Struß, and Thomas Mandl. 2021b. The CLEF-2021 CheckThat! lab on detecting check-worthy claims, previously fact-checked claims, and fake news. In *Proceedings of the 43rd European Conference on Information Retrieval*, ECIR~'21, pages 639–649, Lucca, Italy.

Yixin Nie, Haonan Chen, and Mohit Bansal. 2019. Combining fact extraction and verification with neural semantic matching networks. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial*

Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019, pages 6859–6866. AAAI Press.

Jeppe Nørregaard and Leon Derczynski. 2021. DanFEVER: claim verification dataset for Danish. In *Proceedings of the 23rd Nordic Conference on Computational Linguistics (NoDaLiDa)*, pages 422–428, Reykjavik, Iceland (Online). Linköping University Electronic Press, Sweden.

Javad Nouri and Roman Yangarber. 2016. Modeling language evolution with codes that utilize context and phonetic features. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 136–145, Berlin, Germany. Association for Computational Linguistics.

Wojciech Ostrowski, Arnav Arora, Pepa Atanasova, and Isabelle Augenstein. 2021. Multi-hop fact checking of political claims. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 3892–3898. International Joint Conferences on Artificial Intelligence Organization. Main Track.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Lucia C Passaro, Alessandro Bondielli, and Alessandro Lenci. 2020. UNIPI-NLE at CheckThat! 2020: Approaching Fact Checking from a Sentence Similarity Perspective Through the Lens of Transformers. In *CLEF (Working Notes)*, page 15.

Ayush Patwari, Dan Goldwasser, and Saurabh Bagchi. 2017. TATHYA: A multi-classifier system for detecting check-worthy statements in political debates. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, CIKM 2017, Singapore, November 06 - 10, 2017*, pages 2259–2262. ACM.

Thomas Pellissier Tanon, Gerhard Weikum, and Fabian Suchanek. 2020. YAGO 4: A Reason-able Knowledge Base. In *The Semantic Web*, Lecture Notes in Computer Science, pages 583–596, Cham. Springer International Publishing.

Fabio Petroni, Aleksandra Piktus, Angela Fan, Patrick Lewis, Majid Yazdani, Nicola De Cao, James Thorne, Yacine Jernite, Vladimir Karpukhin, Jean Maillard, Vassilis Plachouras, Tim

Rocktäschel, and Sebastian Riedel. 2021. KILT: a benchmark for knowledge intensive language tasks. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2523–2544, Online. Association for Computational Linguistics.

Dean Pomerleau and Delip Rao. 2017. The fake news challenge: Exploring how artificial intelligence technologies could be leveraged to combat fake news. `http://www.fakenewschallenge.org/`.

Kashyap Popat, Subhabrata Mukherjee, Jannik Strötgen, and Gerhard Weikum. 2017. Where the Truth Lies: Explaining the Credibility of Emerging Claims on the Web and Social Media. In *Proceedings of the 26th International Conference on World Wide Web Companion*, WWW '17 Companion, pages 1003–1012, Republic and Canton of Geneva, CHE. International World Wide Web Conferences Steering Committee.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.

Martin Potthast, Sebastian Köpsel, Benno Stein, and Matthias Hagen. 2016. Clickbait detection. In *European Conference on Information Retrieval*.

Ronak Pradeep, Xueguang Ma, Rodrigo Nogueira, and Jimmy Lin. 2021. Scientific claim verification with VerT5erini. In *Proceedings of the 12th International Workshop on Health Text Mining and Information Analysis*, pages 94–103, online. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *arXiv:1910.10683 [cs, stat]*. ArXiv: 1910.10683 version: 3.

Rajkumar Ramamurthy, Prithviraj Ammanabrolu, Kianté Brantley, Jack Hessel, Rafet Sifa, Christian Bauckhage, Hannaneh Hajishirzi, and Yejin Choi. 2022. Is Reinforcement Learning (Not) for Natural Language Processing: Benchmarks, Baselines, and Building Blocks for Natural Language Policy Optimization. *ArXiv preprint*, abs/2210.01241.

Ashish Rana, Pujit Golchha, Roni Juntunen, Andreea Coajă, Ahmed Elzamarany, Chia-Chien Hung, and Simone Paolo Ponzetto. 2022a. Levirank: Limited query expansion with voting integration for document retrieval and ranking. In *CEUR Workshop Proceedings*, pages 3074–3089.

Ashish Rana, Deepanshu Khanna, Tirthankar Ghosal, Muskaan Singh, Harpreet Singh, and Prashant Singh Rana. 2022b. RerrFact: Reduced Evidence Retrieval Representations for Scientific Claim Verification. *ArXiv preprint*, abs/2202.02646.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Stephen E. Robertson, Steve Walker, Susan Jones, Micheline Hancock-Beaulieu, and Mike Gatford. 1994. Okapi at TREC-3. In *Proceedings of the third text REtrieval conference*, volume 500-225 of *NIST special publication*, pages 109–126. National Institute of Standards and Technology (NIST).

Arkadiy Saakyan, Tuhin Chakrabarty, and Smaranda Muresan. 2021. COVID-fact: Fact extraction and verification of real-world claims on COVID-19 pandemic. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2116–2129, Online. Association for Computational Linguistics.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *ArXiv*.

Maarten Sap, Vered Shwartz, Antoine Bosselut, Yejin Choi, and Dan Roth. 2020. Commonsense reasoning for natural language processing. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts*, pages 27–33, 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.

Timo Schick and Hinrich Schütze. 2021a. Exploiting cloze-questions for few-shot text classification and natural language inference. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 255–269, Online. Association for Computational Linguistics.

Timo Schick and Hinrich Schütze. 2021b. It's not just size that matters: Small language models are also few-shot learners. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2339–2352, Online. Association for Computational Linguistics.

Christopher Schröder, Andreas Niekler, and Martin Potthast. 2022. Revisiting Uncertainty-based Query Strategies for Active Learning with Transformers. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2194–2203, Dublin, Ireland. Association for Computational Linguistics.

Tal Schuster, Adam Fisch, and Regina Barzilay. 2021. Get your vitamin C! robust fact verification with contrastive evidence. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 624–643, Online. Association for Computational Linguistics.

Thomas C. Scott-Phillips and Simon Kirby. 2010. Language evolution in the laboratory. *Trends in Cognitive Sciences*, 14(9):411–417.

Thibault Sellam, Dipanjan Das, and Ankur Parikh. 2020. BLEURT: Learning robust metrics for text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7881–7892, Online. Association for Computational Linguistics.

Prithviraj Sen, Marina Danilevsky, Yunyao Li, Siddhartha Brahma, Matthias Boehm, Laura Chiticariu, and Rajasekar Krishnamurthy. 2020. Learning explainable linguistic expressions with neural inductive logic programming for sentence classification. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4211–4221, Online. Association for Computational Linguistics.

Burr Settles. 2009. Active Learning Literature Survey. Technical Report, University of Wisconsin-Madison Department of Computer Sciences. Accepted: 2012-03-15T17:23:56Z.

Burr Settles. 2012. *Active Learning*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Springer International Publishing, Cham.

H. S. Seung, M. Opper, and H. Sompolinsky. 1992. Query by committee. In *Proceedings of the fifth annual workshop on Computational learning theory*, COLT '92, pages 287–294, New York, NY, USA. Association for Computing Machinery.

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

Shaden Shaar, Alex Nikolov, Nikolay Babulkov, Firoj Alam, Alberto Barrón-Cedeño, Tamer Elsayed, Maram Hasanain, Reem Suwaileh, Fatima Haouari, Giovanni Da San Martino, and Preslav Nakov. 2020b. Overview of checkthat! 2020 english: Automatic identification and verification of claims in social media. In *Conference and Labs of the Evaluation Forum*.

Baoxu Shi and Tim Weninger. 2016. Discriminative predicate path mining for fact checking in knowledge graphs. *Know.-Based Syst.*, 104(C):123–133.

Kai Shu, Amy Sliva, Suhang Wang, Jiliang Tang, and Huan Liu. 2017. Fake news detection on social media: A data mining perspective. *SIGKDD Explor. Newsl.*, 19(1):22–36.

Jake Snell, Kevin Swersky, and Richard S. Zemel. 2017. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 4077–4087.

Dominik Stammbach and Elliott Ash. 2020. e-fever: Explanations and summaries forautomated fact checking. In *Conference for Truth and Trust Online*.

Shane Storks, Qiaozi Gao, and Joyce Y. Chai. 2019. Commonsense Reasoning for Natural Language Understanding: A Survey of Benchmarks, Resources, and Approaches. *arXiv:1904.01172 [cs]*. ArXiv: 1904.01172 version: 1.

Zafar Habeeb Syed, Michael Röder, and Axel-Cyrille Ngonga Ngomo. 2019. Unsupervised Discovery of Corroborative Paths for Fact Validation. In *The Semantic Web – ISWC 2019*, Lecture Notes in Computer Science, pages 630–646, Cham. Springer International Publishing.

Derek Tam, Rakesh R. Menon, Mohit Bansal, Shashank Srivastava, and Colin Raffel. 2021. Improving and Simplifying Pattern Exploiting Training. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4980–4991, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Marta Tatu and Dan Moldovan. 2007. COGEX at RTE 3. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pages 22–27, Prague. Association for Computational Linguistics.

James Thorne and Andreas Vlachos. 2018. Automated fact checking: Task formulations, methods and future directions. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3346–3359, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018a. FEVER: a large-scale dataset for fact extraction and VERification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 809–819, New Orleans, Louisiana. Association for Computational Linguistics.

James Thorne, Andreas Vlachos, Oana Cocarascu, Christos Christodoulopoulos, and Arpit Mittal. 2018b. The fact extraction and VERification (FEVER) shared task. In *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*, pages 1–9, Brussels, Belgium. Association for Computational Linguistics.

James Thorne, Andreas Vlachos, Oana Cocarascu, Christos Christodoulopoulos, and Arpit Mittal. 2019. The FEVER2.0 shared task. In *Proceedings of the Second Workshop on Fact Extraction and VERification (FEVER)*, pages 1–6, Hong Kong, China. Association for Computational Linguistics.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas

Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models.

Andreas Vlachos and Sebastian Riedel. 2014. Fact checking: Task definition and dataset construction. In *Proceedings of the ACL 2014 Workshop on Language Technologies and Computational Social Science*, pages 18–22, Baltimore, MD, USA. Association for Computational Linguistics.

David Wadden, Shanchuan Lin, Kyle Lo, Lucy Lu Wang, Madeleine van Zuylen, Arman Cohan, and Hannaneh Hajishirzi. 2020. Fact or fiction: Verifying scientific claims. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7534–7550, Online. Association for Computational Linguistics.

David Wadden, Kyle Lo, Lucy Lu Wang, Arman Cohan, Iz Beltagy, and Hannaneh Hajishirzi. 2022. MultiVerS: Improving scientific claim verification with weak supervision and full-document context. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 61–76, Seattle, United States. Association for Computational Linguistics.

Nancy X. R. Wang, Diwakar Mahajan, Marina Danilevsky, and Sara Rosenthal. 2021. SemEval-2021 task 9: Fact verification and evidence finding for tabular data in scientific documents (SEM-TAB-FACTS). In *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, pages 317–326, Online. Association for Computational Linguistics.

William Yang Wang. 2017. "liar, liar pants on fire": A new benchmark dataset for fake news detection. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 422–426, Vancouver, Canada. Association for Computational Linguistics.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Evan Williams, Paul Rodrigues, and Valerie Novak. 2020. Accenture at CheckThat! 2020: If you say so: Post-hoc fact-checking of claims using transformer-based models. *arXiv:2009.02431 [cs]*. ArXiv: 2009.02431.

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.

Dustin Wright and Isabelle Augenstein. 2020. Claim check-worthiness detection as positive unlabelled learning. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 476–488, Online. Association for Computational Linguistics.

Michelle Yuan, Hsuan-Tien Lin, and Jordan Boyd-Graber. 2020. Cold-start active learning through self-supervised language modeling. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7935–7948, Online. Association for Computational Linguistics.

Weizhe Yuan, Graham Neubig, and Pengfei Liu. 2021. BARTScore: Evaluating Generated Text as Text Generation. In *Advances in Neural Information Processing Systems*, volume 34, pages 27263–27277. Curran Associates, Inc.

Fabio massimo Zanzotto, Marco Pennacchiotti, and Alessandro Moschitti. 2009. A machine learning approach to textual entailment recognition. *Nat. Lang. Eng.*, 15(4):551–582.

Majid Zarharan, Mahsa Ghaderan, Amin Pourdabiri, Zahra Sayedi, Behrouz Minaei-Bidgoli, Sauleh Eetemadi, and Mohammad Taher Pilehvar. 2021. ParsFEVER: a dataset for Farsi fact extraction and verification. In *Proceedings of *SEM 2021: The Tenth Joint Conference on*

Lexical and Computational Semantics, pages 99–104, Online. Association for Computational Linguistics.

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.

Xia Zeng and Arkaitz Zubiaga. 2021. QMUL-SDS at SCIVER: Step-by-step binary classification for scientific claim verification. In *Proceedings of the Second Workshop on Scholarly Document Processing*, pages 116–123, Online. Association for Computational Linguistics.

Xia Zeng and Arkaitz Zubiaga. 2022. Aggregating pairwise semantic differences for few-shot claim verification. *PeerJ Computer Science*, 8:e1137. Publisher: PeerJ Inc.

Xia Zeng and Arkaitz Zubiaga. 2023. Active PETs: Active Data Annotation Prioritisation for Few-Shot Claim Verification with Pattern Exploiting Training. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 190–204, Dubrovnik, Croatia. Association for Computational Linguistics.

Xia Zeng and Arkaitz Zubiaga. 2024. MAPLE: Micro analysis of pairwise language evolution for few-shot claim verification. In *Findings of the Association for Computational Linguistics: EACL 2024*, pages 1177–1196, St. Julian's, Malta. Association for Computational Linguistics.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with BERT. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Zhiwei Zhang, Jiyi Li, Fumiyo Fukumoto, and Yanming Ye. 2021. Abstract, Rationale, Stance: A Joint Model for Scientific Claim Verification. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3580–3586, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Arkaitz Zubiaga, Ahmet Aker, Kalina Bontcheva, Maria Liakata, and Rob Procter. 2018. Detection and resolution of rumours in social media: A survey. *ACM Comput. Surv.*, 51(2).

Appendix A

Preliminary Study: QMUL@SCIVER

A.1 Introduction

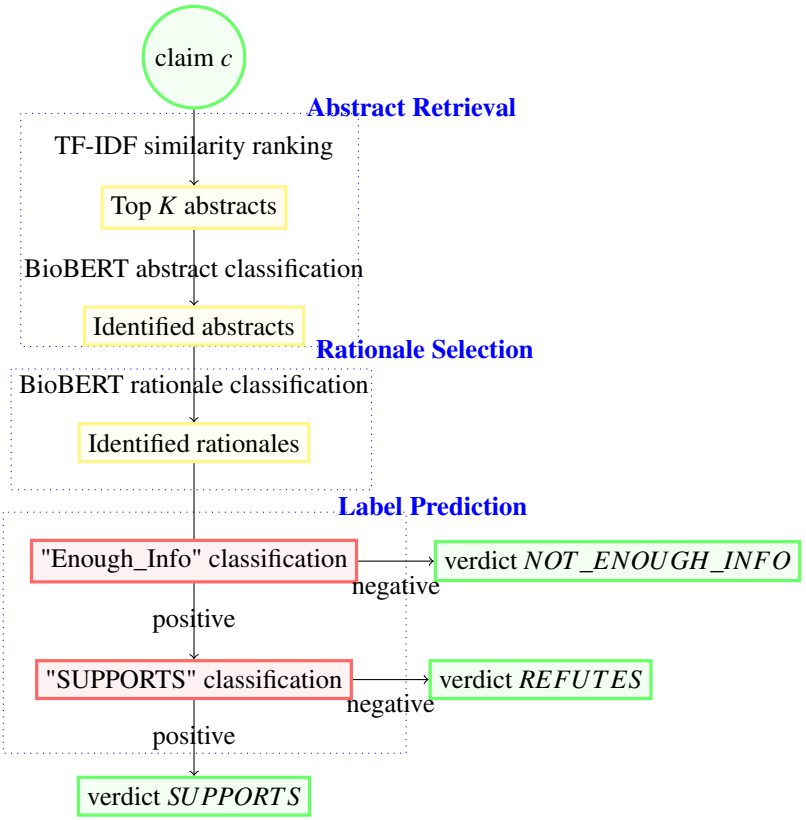

Figure A.1: Overview of our step-by-step binary classification system.

As online content continues to grow at an unprecedented rate, the spread of false information online increases the potential of misleading people and causing harm. Where the volume of

information shared online is difficult to be managed by human fact-checkers, this leads to an increasing demand on automated fact-checking, which is formulated by researchers as 'the assignment of a truth value to a claim made in a particular context'(Vlachos and Riedel, 2014).

Though a body of research focuses on conducting fact-checking in the politics domain, scientific claim verification has also gained increasing interest in the context of the ongoing COVID-19 pandemic. The SCIVER shared task provides a valuable benchmark to build and evaluate systems performing scientific claim verification. Given a scientific claim and a corpus of over 5000 abstracts, the task consists in (i) identifying abstracts relevant to the claim, (ii) delving into the abstracts to select evidence sentences relevant to the claim, and (iii) subsequently predicting claim veracity.

This chapter presents and analyses team QMUL-SDS's participation in the SCIVER shared task. In particular, we explore creative approaches of solving the challenge with limited resources. Figure A.1 provides an overview of our system. Given claim c, our system first retrieves top K TF-IDF similarity abstracts out of the corpus, then uses a BioBERT binary classifier to further identify desired abstracts on top of that. With retrieved abstracts, our system then uses another BioBERT binary classifier to select rationales. We finally do label prediction in a two-step fashion, i.e. first make verdicts on "ENOUGH_INFO" or not and, if positive, then make verdicts on "SUPPORTS" or not. While many other systems make use of external datasets, e.g. FEVER (Thorne et al., 2018a), our system focuses on efficient use of the SciFact dataset (Wadden et al., 2020). Furthermore, in the interest of keeping the efficiency of our system, we limit our model choices to the size of RoBERTa-large (Liu et al., 2019), ruling out for example GPT-3 (Brown et al., 2020) and T5 (Raffel et al., 2020), which were used in other participating systems. More specifically, our system mainly uses RoBERTa (Liu et al., 2019) and BioBERT (Lee et al., 2020). The latter is pre-trained on biomedical text and therefore is very close to our target domain. With improved pipeline design, our system shows competitive performance with limited computing resources, achieving the 6th position in the task and ranked 4th when distinct teams are considered at the time. [1]

[1] Code and models are available here.

A.2 Related Work

Several approaches have been proposed to perform scientific claim verification in the three-step settings proposed in SCIVER.

Upon publication of the SciFact dataset (Wadden et al., 2020), the authors introduced VERISCI as a baseline system. It is a pipeline with three modules: abstract retrieval, rationale selection and label prediction. The abstract retrieval module returns the top K highest-ranked abstracts determined by the TF-IDF similarity between each abstract and the claim at hand. The rationale selection module trains a RoBERTa-large model to compute relevance scores with a sigmoid function and then selects sentences whose relevance scores are higher than the threshold T. The label prediction module trains a RoBERTa-large model to do three-way classification regarding sentence-pairs, where the candidate labels are "SUPPORTS", "REFUTES" and "NOT_ENOUGH_INFO". Empirically the system set the K value to 3 and the T value to 0.5. Due to its inspiring design, reasonable performance and good efficiency, in this chapter we take VERISCI system as our baseline.

After the publication of the SciFact dataset, several approaches have been published, some of which chose to participate in the SCIVER shared task. We next discuss the top 3 ranked entries. The VERT5ERINI system (Pradeep et al., 2021) ranked 1st on the leaderboard. This system first retrieves a shortlist of top 20 abstracts by using the BM25 ranking score (Robertson et al., 1994), which is then fed into a T5 model to rerank and retrieve the top 3 abstracts; it then trains a T5 model to calculate relevance scores for each sentence, on which a threshold of 0.999 is applied to select rationales; it finally trains a T5 model to do three-way classification for predicting labels. This system has demonstrated the performance advantages of using T5, a model that is substantially bigger than other language models.

The ParagraphJoint system (Li et al., 2021) ranked 2nd on the leaderboard. It first uses BioSentVec (Chen et al., 2019a) to retrieve the top K abstracts and then jointly trains a RoBERTa-large model to do rationale selection and label prediction in a multi-task learning setting. The system is first trained on the FEVER dataset and then trained on SciFact dataset. Its application of multi-task learning techniques proved to be very successful and inspires further research in this direction.

The team who ranked 3rd on the leaderboard, Law & Econ (Stammbach and Ash, 2020), fine-tuned their e-FEVER system on SciFact dataset, which requires usage of GPT-3 and training on

FEVER dataset. Despite the big difference on model sizes, our system achieves close performance to the e-FEVER system on the leaderboard.

A.3 Approach

Following the convention of automated fact-checking systems (Thorne et al., 2018a) and the VERISCI baseline system, we explore novel ways of tackling the challenge by handling the three subtasks: abstract retrieval, rationale selection and label prediction.

A.3.1 Abstract Retrieval

Abstract retrieval is the task of retrieving relevant abstracts that can support the prediction of a claim's veracity. Inspired by the baseline system, which retrieves the top K $(K = 3)$ abstracts with the highest TF-IDF similarity to the claim, initially we attempted a similar method with a state-of-the-art similarity metric, i.e., BERTscore (Zhang et al., 2020). It computes token similarity using BERT-based contextual embeddings. However, the results we achieved were not satisfactory and was ruled out in subsequent experiments.

Instead of completely relying on available metrics, we investigated performing abstract retrieval in a supervised manner. In contrast to previous work (Pradeep et al., 2021) which performed reranking, we formulate it as a binary classification problem. We first empirically limit the corpus to the top 30 abstracts with highest TF-IDF similarity to the claim. We fine-tuned a BioBERT model (Lee et al., 2020) with a linear classification head, which we name as the BioBERT classifier thereafter, to do binary classification on the top 30 TF-IDF abstracts, i.e. predicting whether the abstract at hand is correctly identified for the claim at hand given the pairwise input <claim c, title t of the abstract>. Due to the input length limits of BERT models, we only use the title of the abstract at this stage, assuming that the title represents a good summary of the abstract.

A.3.2 Rationale Selection

Rationale Selection is the task of selecting rationale sentences out of the retrieved abstracts. To avoid manually tuning the threshold on various settings like the baseline system, we address the problem as a binary classification task in a very similar manner to the last step. We continued training the BioBERT classifier inherited from the abstract retrieval step to do rationale selection, i.e. making binary predictions on whether the sentence at hand is correctly identified for the claim

at hand given sentence pair <claim *c*, sentence *s*>. As our classifier model only outputs binary predictions with its linear head on individual sentence pair cases, there is no need to apply various ranking thresholds. Aiming to achieve better overall pipeline performance, our models are trained on abstracts retrieved in the first step, rather than oracle abstracts.

A.3.3 Label Prediction

Label prediction is the task of predicting the veracity label given the target claim and rationale sentences selected in the preceding step of the pipeline. A good selection of relevant abstracts and rationales therefore is vital in the capacity of the veracity label prediction system.

The baseline system we initially implemented trained a RoBERTa-large model to do three-way classification into one of "NOT_ENOUGH_INFO", "SUPPORTS" and "REFUTES". We observed that, while the model was in general fairly accurate, it performed poorly in predicting the "REFUTES" class due to the scarcity of training data pertaining to this class. However, it is known that claims belonging to the "REFUTES" class are particularly difficult to collect, and that automated fact-checking datasets tend to create them synthetically by manually mutating naturally occurring claims originally pertaining to the "SUPPORTS" class (Thorne et al., 2018a; Wadden et al., 2020; Sathe et al., 2020). With the aim of improving model performance on this class without using extra data, we try to decrease wrong predictions accumulated by wrong predictions on the other labels. For instance, the model may predict a claim to be "NOT_ENOUGH_INFO" while it should be "REFUTES", which makes it a false positive for the "NOT_ENOUGH_INFO" class and a true negative for the "REFUTES" class. If the model has better performance on the "NOT_ENOUGH_INFO" predictions, it would in turn help the performance on the "REFUTES" class.

Hence, we explore label prediction within a two-step setting. First, we merge claims from the "SUPPORTS" and "REFUTES" classes as "ENOUGH_INFO". With this altered dataset, we train a RoBERTa-large model as a neutral detector to do binary classification into "ENOUGH_INFO" or "NOT_ENOUGH_INFO". Second, we merge data from "NOT_ENOUGH_INFO" and "REFUTES" to be "NOT_SUPPORTS" and train another RoBERTa-large model as a support detector to do binary classification on "SUPPORTS" or "NOT_SUPPORTS". Finally, when doing predictions, we first use the neutral detector to predict "ENOUGH_INFO" or "NOT_ENOUGH_INFO" and only if the first prediction is "ENOUGH_INFO" we use the support detector to predict "SUPPORTS" or "NOT_SUPPORTS". We take "NOT_SUPPORTS" instances as equivalent to

"REFUTES" instances in the three-way classification.

A.4 Results

We perform various experiments on the SciFact dataset to identify the best models and techniques to be submitted to the task. Unless explicitly specified, models are trained on the SciFact's train set and evaluated on the SciFact's dev set.

A.4.1 Abstract Retrieval

We limit the candidate abstracts to the top 30 with the highest TF-IDF similarity scores, as this setting achieves a high recall of 91.39%. With our binary classification method, we experimented with BioBERT models that are pre-trained on close domain texts (Lee et al., 2020). To explore the potentials of adapting pre-trained language models to the current settings, we also conducted task adaptive pre-training (Gururangan et al., 2020) on the SciFact corpus with BioBERT-base for 50 epochs with batch size 1, which leads to a final perplexity of 2.68. This parameter choice is made primarily based on our limited time and computational resources for the SCIVER shared task participation. Further extensive exploration may lead to interesting results. This model is denoted as BioBERT-base*.

Table A.1 reports performance of the baseline, BioBERT-base, BioBERT-base* and BioBERT-large models on abstract retrieval. The baseline directly retrieves the top 3 abstracts with highest TF-IDF similarity, which is also the method used in the VERISCI system (Wadden et al., 2020). We also report abstract level pipeline performance with baseline rationale selector and baseline label predictor to demonstrate its substantial impact on pipeline performance.

Our method achieves noticeable improvements over the baseline by largely decreasing the false positive rate. More specifically, BioBERT-base has the highest precision score, BioBERT-base* has highest F1 score and BioBERT-large has the highest recall score. With increased model size, BioBERT-large has gained significant improvements on recall but suffers with a precision drop compared to BioBERT-base and BioBERT-base*, which may suggest model underfitting. Overall our approach leads to an approximate 10% increase over the baseline approach on abstract level downstream performance.

Abstract Retrieval			
Method	P	R	F1
Baseline	16.22	69.86	26.33
BioBERT-base	**83.23**	64.11	72.43
BioBERT-base*	81.61	67.94	**74.15**
BioBERT-large	62.75	**74.16**	67.98
Downstream Performance			
Abstract Level Label Only			
Method	P	R	F1
Baseline	56.42	48.33	52.06
BioBERT-base	84.30	48.80	61.82
BioBERT-base*	**84.92**	51.20	**63.88**
BioBERT-large	79.71	**52.63**	63.40
Abstract Level Label + Rationale			
Method	P	R	F1
Baseline	54.19	46.41	50.00
BioBERT-base	81.82	47.37	60.00
BioBERT-base*	**82.54**	49.76	**62.09**
BioBERT-large	76.81	**50.72**	61.10

Table A.1: Comparison of abstract retrieval methods on the dev set of SciFact.

A.4.2 Rationale Selection

In order to improve the overall design of the system, we trained our rationale selection models with abstracts retrieved by our abstract retrieval module rather than oracle abstracts. We use abstracts retrieved by BioBERT-large due to its highest recall score. In this step, we experiment with our binary classification approach to identify rationale sentences from retrieved abstracts for the claim at hand. Given a sentence-pair <claim c, sentence s>, the model, which was trained to do abstract selection in last step, is now trained to predict whether the sentence at hand is correctly identified for the claim at hand.

Table A.2 reports results of the baseline, BioBERT-base, BioBERT-base* and BioBERT-large models on rationale selection. We also present sentence level pipeline performance with oracle

Sentence Selection			
Method	P	R	F1
Baseline	64.99	**70.49**	67.63
BioBERT-base	**77.97**	62.84	**69.59**
BioBERT-base*	74.38	65.03	69.39
BioBERT-large	77.08	63.39	69.57
Downstream Performance			
Sentence Level Selection Only			
Method	P	R	F1
Baseline	74.48	**59.02**	65.85
BioBERT-base	**83.81**	56.56	67.54
BioBERT-base*	80.84	57.65	67.30
BioBERT-large	80.75	58.47	**67.83**
Sentence Level Selection + Label			
Method	P	R	F1
Baseline	66.90	53.01	59.15
BioBERT-base	**74.90**	50.55	60.36
BioBERT-base*	72.41	51.64	60.29
BioBERT-large	72.08	**52.19**	**60.54**

Table A.2: Comparison of rationale selection methods on the dev set of SciFact.

citepd abstracts [2] and baseline label predictor.

Our method leads to an increase in precision score, a small decrease in recall score and a small increase in F1 score. Interestingly, the three BioBERT variants don't show clear performance differences, despite substantial differences in model sizes. A small improvement on downstream sentence-level performance is achieved overall.

[2]It includes abstracts that are of "SUPPORTS", "REFUTES" and "NOT_ENOUGH_INFO" relations to the claims' veracity. It is also referred as oracle abstracts with NOT_ENOUGH_INFO (NEI) setting in SciFact dataset paper.

A.4.3 Label Prediction

For label prediction, we use the two-step approach that leverages RoBERTa-large as described in §A.3.3. This approach is denoted as TWO-STEP thereafter. Table A.3 reports performance results for the label prediction task with oracle citepd abstracts and oracle rationales. The baseline is the RoBERTa-large three-way classifier used on VERISCI. Our TWO-STEP method leads to a 4% increase in accuracy, macro-F1 and weighted-F1 over the baseline. We further present confusion matrices for each system for analysis, where R stands for "REFUTES", N stands for "NOT_ENOUGH_INFO" and S stands for "SUPPORTS". As the confusion matrix shows, our method successfully improves the overall predictions on the "REFUTES" class without leveraging extra data.

Furthermore, Table A.4 reports results on the abstract-level label prediction with various settings of upstream modules. Interestingly, both methods show noticeably decreased performance when given an evidence of lower quality. From the oracle evidence to the evidence retrieved by our system, the baseline module's F1 performance dropped by 19.70% and the TWO-STEP module dropped by 20.26% in absolute values; from the oracle evidence to the evidence retrieved by the baseline system, the baseline module's F1 score dropped by 30.14% and the TWO-STEP module dropped by 37.26% in absolute values.

Despite that, our TWO-STEP method always outperforms the baseline method when given improved evidence. Its F1 score is 2.02% - 2.58% higher than the baseline on improved evidence retrieval settings. When given oracle citepd abstracts and oracle rationales, our method achieves 84.78% F1 score.

A.4.4 Full Pipeline

Table A.5 reports full pipeline performance on the SciFact dev set. The baseline is the VERISCI system. We compare pipeline systems with different evidence retrieval models, i.e., BioBERT-base, BioBERT-base* and BioBERT-large, combined with the two-step label predictor using RoBERTa-large.

Overall our system achieves substantial improvements over the baseline. Across the evaluation metrics, our precision scores are 15.75%-23.37% higher than the baseline system, recall scores are 3.82%-14.21% higher and F1 scores are 10.11%-16.08% higher than the baseline in terms of absolute values. Interestingly, BioBERT-base obtains the highest precision score, BioBERT-base*

Label Prediction Performance			
Method	Accuracy	Macro-F1	Weighted-F1
Baseline	81.93	80.19	81.85
TWO-STEP	**85.98**	**84.69**	**85.84**

Confusion Matrix of Baseline			
	R	N	S
R	47	17	7
N	6	104	2
S	8	18	112

Confusion Matrix of TWO-STEP			
	R	N	S
R	53	7	11
N	2	107	3
S	12	10	116

Table A.3: Comparison of label prediction methods with oracle citepd abstracts and oracle rationales.

the highest recall score and BioBERT-large the highest F1 for most of metrics.

Table A.6 compares full pipeline performance on SciFact test set with models trained on the combination of SciFact train set and dev set. OURSYSTEM uses BioBERT-large for abstract retrieval and rationale selection with two-step label prediction, all trained on trained set and dev set. We used BioBERT-large evidence selector and two-step label predictor as our system due to its overall best performance. This submission ranked No. 6 on the leaderboard.

A.5 Discussion and Future Work

Our intuitive step-by-step binary classification system achieves substantial improvements over the baseline without demanding additional data or extra large models.

An improved evidence retrieval module has made the main contributions to the performance boost. Our system makes an effort to improve the abstract retrieval module after applying a scalable traditional information retrieval weighting scheme, TF-IDF. Instead of handling it as a re-ranking task and manually selecting thresholds (Pradeep et al., 2021), we formulate it as a

Oracle Abstract + Oracle Rationale			
Method	P	R	F1
Baseline	**90.75**	75.12	82.20
TWO-STEP	88.54	**81.33**	**84.78**
OurSystem Abstract + OurSystem Rationale			
Method	P	R	F1
Baseline	**76.92**	52.63	62.50
TWO-STEP	73.62	**57.42**	**64.52**
Baseline Abstract + Baseline Rationale			
Method	P	R	F1
Baseline	**56.42**	48.32	**52.06**
TWO-STEP	43.31	**52.63**	47.52

Table A.4: Comparison of label prediction methods with various upstream modules.

binary classification task, which makes better use of the available training data and decreases the false positive rate effectively. When applying a similar approach to rationale selection, our model, which is only trained on the SciFact dataset, still achieves improvements over the baseline model, which makes use of the FEVER dataset first. Furthermore, our model is less dependent on parameters than other systems, which is ideal in practical settings where one would like to apply the model on new datasets without having to find the best parameters for the dataset at hand.

In addition, our TWO-STEP label prediction module also makes positive contributions to overall improvements. The difference on the label prediction performance is very noticeable on different upstream settings. Unsurprisingly, both methods have the best performance with F1 scores higher than 80% on the oracle setting, which is the closest to their training data. Interestingly, this performance fluctuation leads to the following observation: a label prediction module that has better performance on the oracle evidence doesn't necessarily have better performance when given the incorrect evidence. Regarding our TWO-STEP label prediction method, it shows that our neutral detector is not robust enough on the pipeline setting. One possible solution is to train it on evidence retrieved by previous modules rather than on the oracle evidence so that it learns to optimise for the pipeline setting.

Nevertheless, this problem is inevitable for a pipeline system that has multiple machine learning modules, as errors in each of the modules will accumulate throughout the pipeline. A better system design is desired such that it tackles the challenge in a more systematic way. A promising approach is to train a model to learn three subtasks in a multitask learning manner so that it may optimise for better overall performance.

A.6 Summary

In this chapter, we have proposed a novel step-by-step binary classification approach for the SCIVER shared task. Our submission achieved an F1 score of 55.35% on the test set, ranking 6th among all the submissions and 4th among all the teams. We show that (1) concerning evidence retrieval, a classification based approach is better than a ranking based approach with manual thresholds; (2) two-step binary label prediction has better performance than three-way label prediction with limited training data; (3) a more systematic design of automated fact-checking system is desired.

Label Only			
System	P	R	F1
Baseline	56.42	48.33	52.06
BioBERT-base + TWO-STEP	**79.56**	52.15	63.00
BioBERT-base* + TWO-STEP	78.91	55.50	**65.17**
BioBERT-large + TWO-STEP	73.62	**57.42**	64.52
Label+Rationale			
System	P	R	F1
Baseline	54.19	46.41	50.00
BioBERT-base + TWO-STEP	**75.91**	49.76	60.11
BioBERT-base* + TWO-STEP	73.47	51.67	60.67
BioBERT-large + TWO-STEP	69.94	**54.55**	**61.29**
Selection Only			
System	P	R	F1
Baseline	54.27	43.44	48.25
BioBERT-base + TWO-STEP	**77.64**	52.19	62.42
BioBERT-base* + TWO-STEP	72.00	54.10	61.78
BioBERT-large + TWO-STEP	72.76	**57.65**	**64.33**
Selection+Label			
System	P	R	F1
Baseline	48.46	38.80	43.10
BioBERT-base + TWO-STEP	**68.29**	45.90	54.90
BioBERT-base* + TWO-STEP	64.00	48.09	54.92
BioBERT-large + TWO-STEP	64.83	**51.37**	**57.32**

Table A.5: Comparison of full pipeline performance on the dev set of SciFact.

Label Only			
System	P	R	F1
Baseline	47.51	47.30	47.40
OURSYSTEM	**74.32**	**49.55**	**59.46**
Label+Rationale			
System	P	R	F1
Baseline	46.61	46.40	46.50
OURSYSTEM	**72.97**	**48.65**	**58.38**
Selection Only			
System	P	R	F1
Baseline	44.99	47.30	46.11
OURSYSTEM	**81.58**	**58.65**	**68.24**
Selection+Label			
System	P	R	F1
Baseline	38.56	40.54	39.53
OURSYSTEM	**66.17**	**47.57**	**55.35**

Table A.6: Full pipeline performance on SciFact's test set.

Appendix B

Related Tasks

B.1 Claim Detection

Claim detection plays a crucial role in automated fact-checking systems as all other components need to rely on the output of this stage. It aims to relief the burden of identifying claims for fact-checkers and help them by minimising the volume of online content they need to deal with.

B.1.1 Approaches

The claim detection component is responsible for selecting claims that need to go through the rest of the fact-checking pipeline due to needing to be checked, i.e. needing verification. For instance, a factual statement such as "He voted against the first gulf war" can be deemed a claim that should be fact-checked. In contrast, a piece of opinion such as "I think it's time to talk about the future" is not a claim that should be fact-checked (Hassan et al., 2017a).

Going further, one can also distinguish between check-worthy and non-check-worthy claims (Nakov et al., 2021b). For example, one could argue that "the government invested more than 10 billion last year in education" is a claim that is worthy of fact-checking, whereas a claim such as "my friend had a coffee this morning for breakfast" may not be worthy of fact-checking.

Researchers typically formulate the problem as one having a set of sentences as input (e.g. originating from a debate or conversation), and is tackled as a classification task, where a binary decision is made on whether each input sentence constitutes a claim or not, or a ranking task, where input sentences are ranked by check-worthiness, to prioritize most check-worthy claims.

B.1.2 Datasets

In recent studies, several datasets were built with the purpose of enabling training machine learning models to predict check-worthy claims, as shown in Table B.1. The vast majority of datasets cover sentences pertaining to the political domain, as a result of events that synchronously occur with the US elections. In contrast, the CheckThat! Lab released English and Arabic datasets that contain a small number of instances related to COVID-19 in early 2020.

Table B.1: Check-worthiness claim detection Datasets

Name	Size	Annotation type	Language
ClaimBuster (Arslan et al., 2020)	23,533 Sentences	Manual	English
CW-USPD-2016 (Gencheva et al., 2017)	5,415 Sentences	From Existing Annotation	English
TATHYA (Patwari et al., 2017)	15,735 Sentences	From Existing Annotation	English
Konstantinovskiy et al., (Konstantinovskiy et al., 2021)	5,571 Sentences	Based on the annotation scheme, sentences labelled into 7 categories then grouped into 2 categories	English
CT-CWC-18 (Atanasova et al., 2018)	8,946 (En),7,254 (Ar) sentences	From Existing Annotation	English/Partially translated to Arabic
CT19-T1 (Atanasova et al., 2019a)	23,500 sentences	From Existing Annotation	English
Shaar et al., (Shaar et al., 2020b)	962 tweets	Manual	English
CT20-AR (Hasanain et al., 2020)	7.5K tweets	Manual	Arabic
TrClaim-19 (Kartal and Kutlu, 2020)	2287 tweets	Manual	Turkish
FactRank (Berendt et al., 2021)	7037 sentences	Manual	Dutch

In addition, publicly available datasets have a variety of sizes. For instance, ClaimBuster (Arslan et al., 2020) and CT19-T1 (Atanasova et al., 2019a) are the largest datasets, while CW-USPD-2016 (Gencheva et al., 2017), CT-CWC-18 (Atanasova et al., 2018), CT20-AR (Hasanain et al., 2020), and FactRank (Berendt et al., 2021) are a degree of magnitude smaller, followed by other smaller datasets.

Datasets have binary-class for either single-label or multi-labels, depending on the annotation process. The annotation process comes in diverse types. Some datasets are automatically built by collecting claims from fact-checking websites, while other datasets rely on manual annotations given specific definitions of check-worthiness. Crowd-sourcing platform has also been demonstrated to be helpful (Hassan et al., 2015).

Moreover, the majority of datasets are available in the English language as opposed to a smaller number of datasets in the Arabic language. Most of these Arabic datasets are generated from translations of originally English datasets, except for CT20-AR, which is originally Arabic content. In addition, there is one dataset in Dutch and another one in Turkish, while datasets in other languages are not yet available.

B.1.3 Check-Worthy Claim Detection

ClaimBuster is the first automated fact-checking system that consists of integrated components tackling the entire fact-checking pipeline, starting off from the claim detection component. Its claim detection component called "claim spotter" classifies input sentences into one of (1) a factual claim, (2) an unimportant factual claim, or (3) a non-factual claim. This in turn assists fact checkers by prioritising the most check-worthy claims by ranking them based on accuracy measures such as Precision at k (P@K) (Hassan et al., 2017a). To develop this, a multi-class Support Vector Machine (SVM) classifier was built which used features such as bags-of-words, Part-Of-Speech (POS) tags, and Entity Types (ET). The model achieved competitive performance and was considered as the baseline to beat in subsequent works (Hassan et al., 2017a).

Another model called "CNC" (i.e. "Claim/not Claim") (Konstantinovskiy et al., 2021) builds on top of InferSent embeddings (Conneau et al., 2017), combining them with part-of-speech tags and named entities found in texts, which are fed to a Logistic Regression classifier. Authors of CNC had as their main goal the improvement of the recall score achieved by their system, arguing that fact-checkers don't want to miss out any claims (no false negatives) while they can deal with some false positives. While improving in terms of recall, CNC also achieved superior performance in F1 score.

Apart from classic machine learning models, neural networks have also been studied for the claim detection task. For example, in the CheckThat! Lab 2019 shared task, LSTM neural networks and Feed Forward Neural Networks were the most effective models used by the top two participants. Along with the use of neural networks, top participants also showed the usefulness

of context (i.e. surrounding sentences) in improving claim detection performance (Elsayed et al., 2019). The use of context was studied in more detail in another work conducted outside the shared task, in this case by (Atanasova et al., 2019b). They studied the inclusion of context and discourse features along with sentence-level features. They used a Feed-Forward Neural Network (FNN) as the model, which was then evaluated as a ranking task, proving the effectiveness of context and discourse features.

While all aforementioned works focused on English claims, there have also been efforts in other languages. For example, the ClaimRank model (Jaradat et al., 2018) was tested on Arabic claims (translated from claims originally in English). The Arabic claim detection model used Farasa (Abdelali et al., 2016) for tokenization, part-of-speech (POS) tagging, as well as MUSE embeddings. The first experiments on original Arabic data (rather than translated) were conducted in the CheckThat! 2020 shared task. Most participants proposed methods involves fine-tuning pre-trained language models. For instance, the top-performing participant fine-tuned AraBERT v0.1 with neural networks (Williams et al., 2020). Likewise, (Hasanain and Elsayed, 2020) fine-tuned multilingual BERT (mBERT) with different classification models. Another recent effort, called FactRank (Berendt et al., 2021), focused on claim check-worthiness detection for the Dutch language, in this case using a convolutional neural network (CNN) along with Platt scaling for an SVM model and a softmax to obtain the degree of check-worthiness.

B.1.4 Claim Matching

Another task that has recently emerged is claim matching, also referred to as identifying previously fact-checked claims. For a claim spotted in the claim detection component, claim matching consists in determining whether this is a claim that exists in the database and can be resolved by a previous fact-check. The task is formulated as follows: given a check-worthy claim as input, and having a database of previously fact-checked claims, it consists in determining if any of the claims in the database is related to the input; in this case, the new claim would not need fact-checking again, as it was fact-checked in the past. It is normally framed as a ranking task, where claims in the database are ranked based on their similarity to the input claim (Shaar et al., 2020a). This task comes right after the claim detection component, to determine if the claim is new, and can help avoid the need for running the claim validation component for a particular claim when it is found in the database.

There are two released datasets: one based on PolitiFact and the other based on Snopes.

Initial explorations were conducted on using BM25 (Robertson et al., 1994) and BERT-based models respectively as well as building a SVM reranker with features from both approaches (Shaar et al., 2020a). Otherwise, CLEF2020-CheckThat! held a shared task on Verified Claim Retrieval which uses the Snopes dataset. While the baseline system is a simple BM25 system, shared task participants explored various scoring functions, including unsupervised approaches such as Terrier and Elastic Search scores, classic supervised models such as SVM and various BERT-based models (Shaar et al., 2020b). Buster.ai, the winning team, fine-tune a RoBERTa (Liu et al., 2019) model on the task which was first fine-tuned on other fact-checking datasets (Bouziane et al., 2020). Team UNIPI-NLE, achieving close performance to the winning team, performed two cascade fine-tunings on a sentenceBERT (Reimers and Gurevych, 2019) model (Passaro et al., 2020).

B.1.5 Discussion and Challenges

In this section, we discuss current progress in each of the components of the automated fact-checking task, as well as highlight the main open challenges.

Conceptual Definition of Claim The definition of claim check-worthiness is brief (Allein and Moens, 2020). Full Fact describe it as "an assertion about the world that can be checked". In contrast, (Konstantinovskiy et al., 2021) mentioned this definition is not enough to decide whether this claim is worthy for check or not. Similarly, (Berendt et al., 2021) declared that not every factual claim will be verified by fact checkers.

Narrow Domains Claims in the political domain are dominating the interest of journalists and researchers, as can be seen in existing datasets. As an example, (Wright and Augenstein, 2020) investigated the development of a claim check-worthiness detection method that would consistently perform over different domains, in this case rumours on Twitter, Wikipedia citations, and political speeches. However, the method showed important challenges in trying to perform well across domains. Recent research in claim detection has expanded to focus on health claims too, particularly concerning the COVID-19 pandemic.

Annotation Issues Labelling of sentences as claims or non-claims is generally done manually by non-experts (see Table B.1). An alternative to this is to derive labels from previously fact-checked claims collected from fact-checking websites. The main caveat of this approach is that fact-checking websites only list claims, rather than non-claims, which means that one needs to

develop models that only leveraged instances of the positive class, i.e. positive unlabelled learning (Wright and Augenstein, 2020) .

Imbalanced Datasets The majority of datasets are imbalanced where not check-worthy claims outnumber check-worthy claims. While this is possibly due to the nature of the task, existing models can have a tendency to overfit due to this imbalance, which calls for more research to tackle the problem. For example, in the CheckThat! Lab 2020, (Williams et al., 2020) attempted to mitigate the problem of overfitting by resampling the larger number of positive instances that were augmenting data through translation between Arabic and English (Williams et al., 2020).

B.2 Other Related Tasks

There are some other popular tasks in natural language processing which are also related to the accuracy, verifiability and credibility of information, which we briefly discuss next as topics recommended for further reading:

Fake News Detection It is the task of determining whether a news article on the web is accurate or not (Shu et al., 2017). Proposed classification approaches are typically centred on shallow features of the articles: n-grams, characters, stop words, part-of-speech tags, readability scores, term frequency, etc. Some more advanced approaches use additional metadata. However, these approaches are more likely to merely capture patterns of different article styles, rather than to sensibly distinguish reliable and unreliable articles (Hanselowski, 2020).

Rumour Detection It is the task of identifying unverified reports circulating on social media. Predictions are typically made on language subjectivity and metadata on social media (Zubiaga et al., 2018). Despite the relevance of these features, the truth value of a claim does not directly depend on these features.

Clickbait Detection Being considerably different from automated fact-checking, clickbait detection does not require external evidence. Approaches with relatively shallow linguistic features (Chakraborty et al., 2016; Chen et al., 2015; Potthast et al., 2016) have yielded reasonable performance.

Commonsense Reasoning To perform commonsense reasoning (Storks et al., 2019), the model needs to be able to do reasoning beyond the explicit information given in sentence pairs, which is highly valued in automated fact-checking (Thorne and Vlachos, 2018). As a new frontier of

artificial intelligence, novel studies have investigated learned knowledge in pre-trained language models, commonsense integration from external knowledge bases, symbolic knowledge incorporation, etc. However, these tasks are currently under investigation and the field calls for major breakthroughs. For more information, we refer to a recent survey (Storks et al., 2019) and a tutorial (Sap et al., 2020).

B.3 Summary

In this chapter, we have presented related tasks for claim validation with special focus on claim detection. Substantial progress has been made by applying pre-trained language models through designed pipelines, but numerous open challenges still need further research. Claim Detection faces challenges from conceptual definition, narrow domains, annotation issues and imbalanced datasets. In addition, improvements over datasets quality, system integrity and model intepretability are desired for claim validation.

Appendix C

Additional Results

C.1 Detailed Performance Comparison across Few-Shot Claim Verification Methods

Here we present a detailed numeric performance comparison of the methods discussed, as well as alternative model checkpoints for PET[1] and LLaMA 2[2].[3] Tables C.1, C.2, C.3 and C.4 report on FEVER, cFEVER, SciFact_oracle and SciFact_retrieved dataset configurations respectively.

FEVER		F1		Accuracy	
n-shot	method	mean	std	mean	std
1	Llama-2-7b-chat-hf	0.3776	0.0438	0.4771	0.0439
	Llama-2-13b-chat-hf	0.4351	0.0613	0.5034	0.0506
	Llama-2-70b-chat-hf	0.2617	0.0427	0.3800	0.0258
	MAPLE	0.6155	0.0645	0.6459	0.0506
	PET_microsoft/deberta-base-mnli	0.3394	0.0351	0.3582	0.0293
	PET_microsoft/deberta-large-mnli	0.4978	0.1011	0.5193	0.0877

[1]We report all six model checkpoints used in Active PETs.

[2]We report all three models that have chat capabilities.

[3]When the same prompt we deigned for 7b model is used on 13b and 70b models, the model performance is significantly lower and even fails to yield responses in many cases and vise versa. Hence, the results for 13b and 70b models in this section are generated with a prompt that is slightly different from the one we used for 7b model. The prompt we used here is "Please perform the task of claim verification. Given a claim and a piece of evidence, your goal is to classify them into one of the following classes: 'SUPPORTS', 'REFUTES' and 'NOT_ENOUGH_INFO'. Here are a few examples: Claim: 'train_claim_i' Evidence: 'train_evidences_i' 'train_labels_i'.". The post-process remains the same.

	PET_roberta-large-mnli	0.2158	0.0516	0.2408	0.0670
	PET_textattack/bert-base-uncased-MNLI	0.3731	0.0456	0.4089	0.0278
	PET_textattack/roberta-base-MNLI	0.2190	0.0409	0.3139	0.0383
	PET_yoshitomo-matsubara/bert-large-uncased-mnli	0.4214	0.0480	0.4509	0.0429
	SEED_bert-base-nli-mean-tokens	0.2724	0.0689	0.3748	0.0494
2	Llama-2-7b-chat-hf	0.3827	0.0301	0.4796	0.0314
	Llama-2-13b-chat-hf	0.3929	0.0504	0.4719	0.0393
	Llama-2-70b-chat-hf	0.2745	0.0402	0.3883	0.0256
	MAPLE	0.6514	0.0460	0.6724	0.0379
	PET_microsoft/deberta-base-mnli	0.3773	0.0354	0.3870	0.0374
	PET_microsoft/deberta-large-mnli	0.5897	0.0917	0.6023	0.0843
	PET_roberta-large-mnli	0.2308	0.0463	0.2526	0.0617
	PET_textattack/bert-base-uncased-MNLI	0.4151	0.0372	0.4338	0.0261
	PET_textattack/roberta-base-MNLI	0.2661	0.0408	0.3349	0.0340
	PET_yoshitomo-matsubara/bert-large-uncased-mnli	0.4689	0.0490	0.4904	0.0448
	SEED_bert-base-nli-mean-tokens	0.3935	0.0822	0.4455	0.0667
3	Llama-2-7b-chat-hf	0.3760	0.0321	0.4702	0.0312
	Llama-2-13b-chat-hf	0.3815	0.0371	0.4606	0.0299
	Llama-2-70b-chat-hf	0.2792	0.0379	0.3930	0.0246
	MAPLE	0.6768	0.0448	0.6911	0.0400
	PET_microsoft/deberta-base-mnli	0.3977	0.0327	0.4069	0.0315
	PET_microsoft/deberta-large-mnli	0.6586	0.0768	0.6649	0.0733
	PET_roberta-large-mnli	0.2551	0.0406	0.2682	0.0513
	PET_textattack/bert-base-uncased-MNLI	0.4429	0.0267	0.4524	0.0213
	PET_textattack/roberta-base-MNLI	0.2810	0.0361	0.3389	0.0330
	PET_yoshitomo-matsubara/bert-large-uncased-mnli	0.4999	0.0401	0.5186	0.0367
	SEED_bert-base-nli-mean-tokens	0.4843	0.0714	0.5118	0.0615
4	Llama-2-7b-chat-hf	0.3621	0.0473	0.4562	0.0408

	Llama-2-13b-chat-hf	0.3790	0.0425	0.4598	0.0343
	Llama-2-70b-chat-hf	0.2874	0.0382	0.3988	0.0248
	MAPLE	0.6909	0.0399	0.7019	0.0368
	PET_microsoft/deberta-base-mnli	0.4142	0.0292	0.4203	0.0293
	PET_microsoft/deberta-large-mnli	0.6893	0.0628	0.6943	0.0603
	PET_roberta-large-mnli	0.2786	0.0405	0.2993	0.0517
	PET_textattack/bert-base-uncased-MNLI	0.4623	0.0211	0.4667	0.0186
	PET_textattack/roberta-base-MNLI	0.3000	0.0353	0.3445	0.0326
	PET_yoshitomo-matsubara/bert-large-uncased-mnli	0.5191	0.0364	0.5318	0.0326
	SEED_bert-base-nli-mean-tokens	0.5331	0.0619	0.5495	0.0568
5	Llama-2-7b-chat-hf	0.3613	0.0468	0.4472	0.0367
	Llama-2-13b-chat-hf	0.3781	0.0320	0.4592	0.0275
	Llama-2-70b-chat-hf	0.2997	0.0371	0.4074	0.0247
	MAPLE	0.6964	0.0403	0.7058	0.0368
	PET_microsoft/deberta-base-mnli	0.4266	0.0270	0.4320	0.0274
	PET_microsoft/deberta-large-mnli	0.7191	0.0584	0.7237	0.0564
	PET_roberta-large-mnli	0.2941	0.0396	0.3188	0.0443
	PET_textattack/bert-base-uncased-MNLI	0.4699	0.0173	0.4731	0.0153
	PET_textattack/roberta-base-MNLI	0.3064	0.0293	0.3456	0.0293
	PET_yoshitomo-matsubara/bert-large-uncased-mnli	0.5267	0.0358	0.5410	0.0318
	SEED_bert-base-nli-mean-tokens	0.5714	0.0556	0.5821	0.0538

Table C.1: Detailed performance on FEVER. The reported results are mean and standard deviation for F1 and accuracy scores on 100 runs.

cFEVER		F1		Accuracy	
n-shot	method	mean	std	mean	std
1	Llama-2-7b-chat-hf	0.3798	0.0346	0.4184	0.0226
	Llama-2-13b-chat-hf	0.4769	0.0380	0.4831	0.0345

	Llama-2-70b-chat-hf	0.2793	0.0439	0.3620	0.0263
	MAPLE	0.3276	0.0717	0.3622	0.0696
	PET_microsoft/deberta-base-mnli	0.2401	0.0209	0.3072	0.0221
	PET_microsoft/deberta-large-mnli	0.3519	0.0672	0.3795	0.0657
	PET_roberta-large-mnli	0.2828	0.0594	0.3078	0.0555
	PET_textattack/bert-base-uncased-MNLI	0.2721	0.0198	0.3151	0.0159
	PET_textattack/roberta-base-MNLI	0.1850	0.0103	0.3175	0.0166
	PET_yoshitomo-matsubara/bert-large-uncased-mnli	0.3519	0.0382	0.3782	0.0302
	SEED_bert-base-nli-mean-tokens	0.2834	0.0621	0.3640	0.0464
2	Llama-2-7b-chat-hf	0.3541	0.0228	0.4067	0.0180
	Llama-2-13b-chat-hf	0.3745	0.0602	0.4007	0.0390
	Llama-2-70b-chat-hf	0.2481	0.0363	0.3389	0.0209
	MAPLE	0.3700	0.0788	0.3899	0.0748
	PET_microsoft/deberta-base-mnli	0.2574	0.0175	0.3069	0.0215
	PET_microsoft/deberta-large-mnli	0.3958	0.0633	0.4148	0.0581
	PET_roberta-large-mnli	0.3147	0.0615	0.3329	0.0597
	PET_textattack/bert-base-uncased-MNLI	0.2898	0.0172	0.3129	0.0162
	PET_textattack/roberta-base-MNLI	0.1962	0.0159	0.3199	0.0200
	PET_yoshitomo-matsubara/bert-large-uncased-mnli	0.3621	0.0364	0.3846	0.0268
	SEED_bert-base-nli-mean-tokens	0.3574	0.0621	0.4020	0.0538
3	Llama-2-7b-chat-hf	0.3638	0.0287	0.4041	0.0188
	Llama-2-13b-chat-hf	0.3866	0.0534	0.4091	0.0359
	Llama-2-70b-chat-hf	0.2515	0.0333	0.3448	0.0153
	MAPLE	0.3993	0.0678	0.4112	0.0643
	PET_microsoft/deberta-base-mnli	0.2665	0.0179	0.3059	0.0190
	PET_microsoft/deberta-large-mnli	0.4081	0.0601	0.4215	0.0603
	PET_roberta-large-mnli	0.3278	0.0565	0.3448	0.0549
	PET_textattack/bert-base-uncased-MNLI	0.2965	0.0141	0.3107	0.0151
	PET_textattack/roberta-base-MNLI	0.2046	0.0195	0.3196	0.0230

	PET_yoshitomo-matsubara/bert-large-uncased-mnli	0.3675	0.0374	0.3943	0.0242
	SEED_bert-base-nli-mean-tokens	0.3857	0.0550	0.4180	0.0559
4	Llama-2-7b-chat-hf	0.3662	0.0243	0.4001	0.0157
	Llama-2-13b-chat-hf	0.4158	0.0466	0.4284	0.0388
	Llama-2-70b-chat-hf	0.2631	0.0337	0.3514	0.0169
	MAPLE	0.4089	0.0677	0.4181	0.0648
	PET_microsoft/deberta-base-mnli	0.2750	0.0202	0.3105	0.0198
	PET_microsoft/deberta-large-mnli	0.4324	0.0424	0.4456	0.0420
	PET_roberta-large-mnli	0.3504	0.0533	0.3652	0.0487
	PET_textattack/bert-base-uncased-MNLI	0.3033	0.0143	0.3141	0.0139
	PET_textattack/roberta-base-MNLI	0.2109	0.0196	0.3221	0.0209
	PET_yoshitomo-matsubara/bert-large-uncased-mnli	0.3710	0.0338	0.3972	0.0218
	SEED_bert-base-nli-mean-tokens	0.4069	0.0477	0.4344	0.0467
5	Llama-2-7b-chat-hf	0.3709	0.0271	0.3932	0.0191
	Llama-2-13b-chat-hf	0.4473	0.0417	0.4540	0.0367
	Llama-2-70b-chat-hf	0.2752	0.0375	0.3575	0.0182
	MAPLE	0.4208	0.0548	0.4299	0.0520
	PET_microsoft/deberta-base-mnli	0.2838	0.0198	0.3148	0.0215
	PET_microsoft/deberta-large-mnli	0.4488	0.0443	0.4606	0.0431
	PET_roberta-large-mnli	0.3587	0.0497	0.3751	0.0424
	PET_textattack/bert-base-uncased-MNLI	0.3049	0.0132	0.3129	0.0127
	PET_textattack/roberta-base-MNLI	0.2121	0.0189	0.3200	0.0208
	PET_yoshitomo-matsubara/bert-large-uncased-mnli	0.3719	0.0311	0.4001	0.0200
	SEED_bert-base-nli-mean-tokens	0.4164	0.0380	0.4409	0.0371

Table C.2: Detailed performance on cFEVER. The reported results are mean and standard deviation for F1 and accuracy scores on 100 runs.

SciFact_oracle	F1	Accuracy

n-shot	method	mean	std	mean	std
1	Llama-2-7b-chat-hf	0.3746	0.0306	0.4549	0.0295
	Llama-2-13b-chat-hf	0.3722	0.0481	0.4359	0.0375
	Llama-2-70b-chat-hf	0.2502	0.0417	0.3706	0.0233
	MAPLE	0.3938	0.0658	0.4333	0.0604
	PET_microsoft/deberta-base-mnli	0.2459	0.0244	0.3112	0.0121
	PET_microsoft/deberta-large-mnli	0.4467	0.0833	0.4699	0.0735
	PET_roberta-large-mnli	0.2514	0.0537	0.2747	0.0569
	PET_textattack/bert-base-uncased-MNLI	0.3696	0.0435	0.4059	0.0314
	PET_textattack/roberta-base-MNLI	0.2352	0.0273	0.3338	0.0301
	PET_yoshitomo-matsubara/bert-large-uncased-mnli	0.3078	0.0255	0.3312	0.0257
	SEED_bert-base-nli-mean-tokens	0.2996	0.0634	0.3757	0.0489
2	Llama-2-7b-chat-hf	0.3812	0.0233	0.4678	0.0237
	Llama-2-13b-chat-hf	0.3489	0.0382	0.4180	0.0313
	Llama-2-70b-chat-hf	0.2614	0.0329	0.3698	0.0176
	MAPLE	0.4263	0.0571	0.4493	0.0575
	PET_microsoft/deberta-base-mnli	0.2686	0.0170	0.3152	0.0120
	PET_microsoft/deberta-large-mnli	0.5099	0.0772	0.5265	0.0673
	PET_roberta-large-mnli	0.2824	0.0503	0.3014	0.0569
	PET_textattack/bert-base-uncased-MNLI	0.3973	0.0337	0.4218	0.0266
	PET_textattack/roberta-base-MNLI	0.2534	0.0280	0.3378	0.0304
	PET_yoshitomo-matsubara/bert-large-uncased-mnli	0.3068	0.0279	0.3401	0.0196
	SEED_bert-base-nli-mean-tokens	0.3552	0.0648	0.3937	0.0600
3	Llama-2-7b-chat-hf	0.3998	0.0377	0.4662	0.0281
	Llama-2-13b-chat-hf	0.3475	0.0395	0.4112	0.0315
	Llama-2-70b-chat-hf	0.2739	0.0377	0.3753	0.0227
	MAPLE	0.4487	0.0402	0.4655	0.0384
	PET_microsoft/deberta-base-mnli	0.2841	0.0163	0.3237	0.0120
	PET_microsoft/deberta-large-mnli	0.5508	0.0722	0.5639	0.0637

	PET_roberta-large-mnli	0.2936	0.0448	0.3159	0.0516
	PET_textattack/bert-base-uncased-MNLI	0.4153	0.0253	0.4312	0.0197
	PET_textattack/roberta-base-MNLI	0.2633	0.0256	0.3372	0.0276
	PET_yoshitomo-matsubara/bert-large-uncased-mnli	0.3047	0.0258	0.3427	0.0181
	SEED_bert-base-nli-mean-tokens	0.4007	0.0593	0.4290	0.0593
4	Llama-2-7b-chat-hf	0.4002	0.0420	0.4542	0.0312
	Llama-2-13b-chat-hf	0.3558	0.0365	0.4165	0.0306
	Llama-2-70b-chat-hf	0.2939	0.0454	0.3888	0.0277
	MAPLE	0.4520	0.0426	0.4661	0.0405
	PET_microsoft/deberta-base-mnli	0.2932	0.0180	0.3265	0.0132
	PET_microsoft/deberta-large-mnli	0.5698	0.0738	0.5781	0.0677
	PET_roberta-large-mnli	0.2988	0.0540	0.3173	0.0585
	PET_textattack/bert-base-uncased-MNLI	0.4197	0.0220	0.4361	0.0157
	PET_textattack/roberta-base-MNLI	0.2743	0.0263	0.3416	0.0287
	PET_yoshitomo-matsubara/bert-large-uncased-mnli	0.3054	0.0269	0.3461	0.0187
	SEED_bert-base-nli-mean-tokens	0.4289	0.0519	0.4499	0.0503
5	Llama-2-7b-chat-hf	0.3998	0.0463	0.4487	0.0328
	Llama-2-13b-chat-hf	0.3611	0.0348	0.4231	0.0308
	Llama-2-70b-chat-hf	0.2840	0.0709	0.3873	0.0370
	MAPLE	0.4554	0.0356	0.4675	0.0356
	PET_microsoft/deberta-base-mnli	0.3005	0.0172	0.3312	0.0139
	PET_microsoft/deberta-large-mnli	0.5964	0.0706	0.6045	0.0641
	PET_roberta-large-mnli	0.3087	0.0507	0.3281	0.0558
	PET_textattack/bert-base-uncased-MNLI	0.4252	0.0233	0.4413	0.0147
	PET_textattack/roberta-base-MNLI	0.2780	0.0222	0.3420	0.0249
	PET_yoshitomo-matsubara/bert-large-uncased-mnli	0.3072	0.0274	0.3496	0.0166
	SEED_bert-base-nli-mean-tokens	0.4463	0.0478	0.4645	0.0465

Table C.3: Detailed performance on SciFact_oracle. The reported results are mean and standard deviation for F1 and accuracy scores on 100 runs.

SciFact_retrieved		F1		Accuracy	
n-shot	method	mean	std	mean	std
1	Llama-2-7b-chat-hf	0.3207	0.0299	0.3943	0.0243
	Llama-2-13b-chat-hf	0.3757	0.0380	0.4265	0.0231
	Llama-2-70b-chat-hf	0.3454	0.0598	0.4035	0.0338
	MAPLE	0.4108	0.0878	0.4412	0.0831
	PET_microsoft/deberta-base-mnli	0.2927	0.0341	0.3134	0.0302
	PET_microsoft/deberta-large-mnli	0.3332	0.0525	0.3609	0.0450
	PET_roberta-large-mnli	0.2448	0.0308	0.2830	0.0298
	PET_textattack/bert-base-uncased-MNLI	0.3431	0.0263	0.3661	0.0180
	PET_textattack/roberta-base-MNLI	0.2598	0.0317	0.3491	0.0238
	PET_yoshitomo-matsubara/bert-large-uncased-mnli	0.3162	0.0352	0.3477	0.0215
	SEED_bert-base-nli-mean-tokens	0.2708	0.0470	0.3479	0.0288
2	Llama-2-7b-chat-hf	0.2914	0.0528	0.3586	0.0350
	Llama-2-13b-chat-hf	0.3278	0.0524	0.3925	0.0266
	Llama-2-70b-chat-hf	0.1682	0.0105	0.3338	0.0038
	MAPLE	0.4484	0.0699	0.4654	0.0675
	PET_microsoft/deberta-base-mnli	0.2988	0.0315	0.3147	0.0281
	PET_microsoft/deberta-large-mnli	0.3601	0.0524	0.3834	0.0434
	PET_roberta-large-mnli	0.2576	0.0300	0.2891	0.0281
	PET_textattack/bert-base-uncased-MNLI	0.3514	0.0201	0.3633	0.0179
	PET_textattack/roberta-base-MNLI	0.2944	0.0289	0.3549	0.0267
	PET_yoshitomo-matsubara/bert-large-uncased-mnli	0.3156	0.0333	0.3571	0.0199
	SEED_bert-base-nli-mean-tokens	0.3233	0.0463	0.3623	0.0439
3	Llama-2-7b-chat-hf	0.1775	0.0363	0.3329	0.0056
	Llama-2-13b-chat-hf	0.1788	0.0371	0.3359	0.0104

	Llama-2-70b-chat-hf	0.1667	0.0000	0.3333	0.0000
	MAPLE	0.4768	0.0511	0.4909	0.0464
	PET_microsoft/deberta-base-mnli	0.2963	0.0308	0.3085	0.0249
	PET_microsoft/deberta-large-mnli	0.3599	0.0518	0.3880	0.0419
	PET_roberta-large-mnli	0.2557	0.0266	0.2853	0.0243
	PET_textattack/bert-base-uncased-MNLI	0.3490	0.0212	0.3604	0.0179
	PET_textattack/roberta-base-MNLI	0.3135	0.0251	0.3559	0.0250
	PET_yoshitomo-matsubara/bert-large-uncased-mnli	0.3102	0.0281	0.3580	0.0171
	SEED_bert-base-nli-mean-tokens	0.3530	0.0382	0.3795	0.0367
4	Llama-2-7b-chat-hf	0.1667	0.0000	0.3333	0.0000
	Llama-2-13b-chat-hf	0.1667	0.0000	0.3333	0.0000
	Llama-2-70b-chat-hf	0.1667	0.0000	0.3333	0.0000
	MAPLE	0.4777	0.0449	0.4884	0.0429
	PET_microsoft/deberta-base-mnli	0.3038	0.0278	0.3129	0.0252
	PET_microsoft/deberta-large-mnli	0.3827	0.0494	0.4026	0.0453
	PET_roberta-large-mnli	0.2616	0.0236	0.2862	0.0224
	PET_textattack/bert-base-uncased-MNLI	0.3467	0.0240	0.3611	0.0195
	PET_textattack/roberta-base-MNLI	0.3289	0.0284	0.3611	0.0245
	PET_yoshitomo-matsubara/bert-large-uncased-mnli	0.3083	0.0253	0.3582	0.0173
	SEED_bert-base-nli-mean-tokens	0.3581	0.0383	0.3820	0.0369
5	Llama-2-7b-chat-hf	0.1667	0.0000	0.3333	0.0000
	Llama-2-13b-chat-hf	0.1667	0.0000	0.3333	0.0000
	Llama-2-70b-chat-hf	0.1667	0.0000	0.3333	0.0000
	MAPLE	0.4846	0.0351	0.4941	0.0331
	PET_microsoft/deberta-base-mnli	0.3054	0.0261	0.3163	0.0240
	PET_microsoft/deberta-large-mnli	0.3825	0.0504	0.4043	0.0435
	PET_roberta-large-mnli	0.2575	0.0274	0.2915	0.0225
	PET_textattack/bert-base-uncased-MNLI	0.3467	0.0242	0.3624	0.0197
	PET_textattack/roberta-base-MNLI	0.3348	0.0252	0.3600	0.0226

PET_yoshitomo-matsubara/bert-large-uncased-mnli	0.3066	0.0289	0.3638	0.0165
SEED_bert-base-nli-mean-tokens	0.3726	0.0361	0.3903	0.0367

Table C.4: Detailed performance on SciFact_retrieved. The reported results are mean and standard deviation for F1 and accuracy scores on 100 runs.

C.2 MAPLE Classwise Performance within 5 Shots

Table C.5 presents MAPLE's classwise performance. In general, MAPLE is most capable of distinguishing NOT_ENOUGH_INFO samples from the others and the least capable when dealing with REFUTES samples.

C.3 MAPLE Performance Comparison within 50 Shots

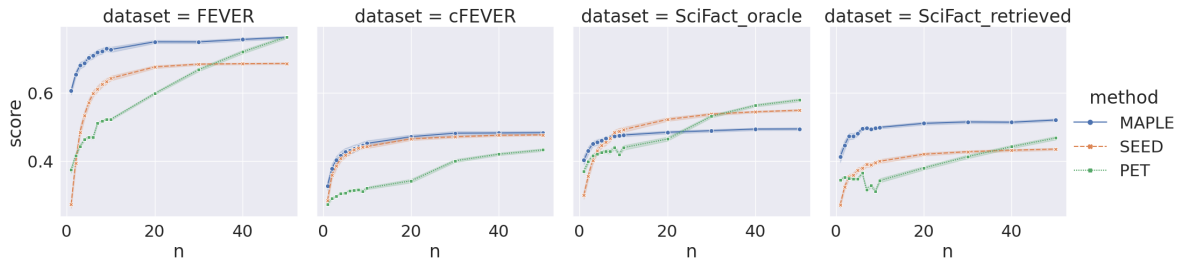

Figure C.1: F1 performance within 50 shots.

Figure C.1 illustrates the F1 results within the 50-shot setting. The experiments are conducted on SEED, PET and MAPLE, as LLaMA 2 imposes high demand on computational budget. MAPLE demonstrates superior performance in three out of four dataset configurations, specifically FEVER, cFEVER, and SciFact_retrieved. Although it is not the top performing approach in the SciFact_oracle setting, it holds the highest position until surpassed by SEED at 8 shots, followed by PET at 30 shots.

On the FEVER dataset, MAPLE achieves significant improvements over the baselines when provided with fewer than 50 shots. MAPLE starts with a very high performance around 0.6 and converges around 20 shots, reaching approximately 0.8. Despite starting from a very low point, SEED learns rapidly within 10 shots and converges around 20 shots with a score below 0.7. PET demonstrates remarkable learning capabilities within 50 shots, as its performance steadily rises to around 0.8.

FEVER						
n-shot	F1(SUPPORTS)		F1(NOT_ENOUGH_INFO)		F1(REFUTES)	
	mean	std	mean	std	mean	std
1	0.4737	0.1665	0.9177	0.1010	0.4550	0.1557
2	0.5144	0.1167	0.9442	0.0270	0.4955	0.1330
3	0.5593	0.1077	0.9531	0.0193	0.5181	0.0972
4	0.5762	0.0938	0.9550	0.0186	0.5416	0.0807
5	0.5821	0.0891	0.9584	0.0157	0.5487	0.0805
cFEVER						
n-shot	F1(SUPPORTS)		F1(NOT_ENOUGH_INFO)		F1(REFUTES)	
	mean	std	mean	std	mean	std
1	0.3333	0.1540	0.3325	0.1679	0.3169	0.1363
2	0.3750	0.1367	0.3810	0.1415	0.3541	0.1191
3	0.4218	0.1159	0.4099	0.1263	0.3663	0.0926
4	0.4162	0.1119	0.4299	0.1154	0.3805	0.0885
5	0.4251	0.1044	0.4538	0.1005	0.3836	0.0773
SciFact_oracle						
n-shot	F1(SUPPORTS)		F1(NOT_ENOUGH_INFO)		F1(REFUTES)	
	mean	std	mean	std	mean	std
1	0.3326	0.1764	0.5141	0.1518	0.3346	0.1568
2	0.3295	0.1326	0.5702	0.1192	0.3794	0.0961
3	0.3780	0.1168	0.5931	0.0741	0.3750	0.0766
4	0.3849	0.1090	0.5882	0.0879	0.3830	0.0737
5	0.3975	0.0992	0.5943	0.0656	0.3744	0.0746
SciFact_retrieved						
n-shot	F1(SUPPORTS)		F1(NOT_ENOUGH_INFO)		F1(REFUTES)	
	mean	std	mean	std	mean	std
1	0.3369	0.1542	0.5438	0.1751	0.3519	0.1525
2	0.3612	0.1199	0.5910	0.1524	0.3930	0.1117
3	0.4030	0.0983	0.6407	0.1045	0.3868	0.0949
4	0.4063	0.0822	0.6409	0.0857	0.3859	0.0922
5	0.3994	0.0867	0.6555	0.0632	0.3989	0.0713

Table C.5: MAPLE Classwise F1 results. The reported results are mean and standard deviation classwise F1 scores for each class on 100 runs.

On the cFEVER dataset, MAPLE remains the best-performing method within 50 shots, although with only a slight margin over SEED. Both MAPLE and SEED exhibit similar performance curves, converging around 20 to 30 shots with scores approaching 0.5. PET shows a different pattern, steadily learning over the range of 50 shots but ending with a lower score compared to the other methods.

On the SciFact_oracle dataset, MAPLE starts strongly but shows limited improvements with more data, converging within 8 shots at approximately 0.48. This may be attributed to the challenging nature of the scientific domain. SEED and PET manage to surpass MAPLE in this case, with SEED converging at 50 shots and achieving a score of around 0.55. PET surpasses MAPLE after being provided with over 20 shots and surpasses SEED after receiving over 30 shots.

On the SciFact_retrieved dataset, unlike in the SciFact_oracle case, MAPLE maintains a clear advantage within 50 shots. MAPLE starts above 0.4 and converges around 20 to 30 shots with a score above 0.5. With retrieved evidence, both SEED and PET experience a performance dip compared to the oracle evidence scenario. SEED also converges around 20 to 30 shots, but with a score above 0.4. PET experiences a dip early on, around 10 shots, dropping to approximately 0.3, despite starting around 0.35. Afterwards, it recovers and reaches above 0.45 at 50 shots, although still lower than MAPLE.

Appendix D

Runtime Reports

D.1 MAPLE with LoRA vs SFT Runtime Comparison

	FEVER	cFEVER	SciFact_oracle	SciFact_retrieved
LoRA runtime (from claim to evidence)	00:50:24	00:39:14	00:05:33	00:16:29
SFT runtime (from claim to evidence)	01:50:52	01:15:14	00:13:23	00:48:21
LoRA runtime (from evidence to claim)	00:50:23	00:39:12	00:05:18	00:16:28
SFT runtime (from evidence to claim)	01:37:58	01:14:39	00:11:41	00:35:12

Table D.1: LoRA vs SFT Runtime comparison. The time format is hours:minutes:seconds.

We present the runtime comparison of LoRA and SFT on performing Seq2seq training on T5-small. While the efficiency gain varies on the given training data, table D.1 shows that significant time savings across all experimented datasets.

D.2 MAPLE Overall Runtime

We present the runtime of MAPLE across four dataset configurations in Table D.2. The experiments were conducted on a High-Performance Compute cluster provided by the university, featuring 8 compute cores, 11G RAM per core, and a single NVIDIA A100 GPU. Seq2seq LoRA training and SemSim transformation were applied to the entire dataset. The LR runtime denotes the execution time for all few-shot experiments outlined in Section 5.2. It's important to note that the runtime is strongly correlated with the size of the unlabelled pool, as well as the length

of claims and evidences. Consequently, it takes a few hours to run for large-scale datasets like FEVER and cFEVER, as well as dataset configurations comprising lengthy instances such as SciFact_retrieved, but considerably less time for SciFact_oracle. For improved efficiency, future work may explore applying the SemSim transformation solely to the sampled few-shot training instances per experiment.

	FEVER	**cFEVER**	**SciFact_oracle**	**SciFact_retrieved**
Seq2Seq runtime (from claim to evidence)	00:50:24	00:39:14	00:05:33	00:16:29
SemSim runtime (from claim to evidence)	00:50:16	00:37:34	00:06:22	00:26:06
Seq2Seq runtime (from evidence to claim)	00:50:23	00:39:12	00:05:18	00:16:28
SemSim runtime (from evidence to claim)	00:49:02	00:37:34	00:05:45	00:23:06
LR runtime	00:00:28	00:00:33	00:00:31	00:00:33
Total runtime	03:20:33	02:34:07	00:23:29	01:22:42

Table D.2: MAPLE runtime on four dataset configurations. The time format is hours:minutes:seconds.

D.3 Active PETs Runtime

We use High Performance Compute cluster supported by the university. Each experiment is run with 8 compute cores, 11G RAM per core and a single NVIDIA A100 GPU. Table D.3 reports the average run time of executing a sampling iteration of 150 unlabelled instances and a training iteration with the sampled data over three datasets. It serves as a good indicator for comparing the efficiency among different active learning methods. As CAL requires an initial labelled set of data, we report the total run time of an iteration of using the random method for 75 instances and an iteration of using CAL method for another 75 instances. Table D.4 further reports the total run time of the best method Active PETs-o on different datasets. The actual run time highly correlates with the size of the unlabelled pool for each datasets.

Our key focus has been on resource-efficiency and performance, with a lesser focus on runtime, hence there can be room for optimisation in future work, including: (1) optimising the code e.g. through parallelisation of the ensembled models which are now run sequentially, (2) using DL optimisation libraries such as deepspeed, and (3) using dynamic step sizes to reduce the number of iterations, e.g. increase step size if initial iterations lead to balanced samples. In a real-world, deployed scenario, one would also need to account for the time needed by humans to perform the

annotation (in our case simulated).

	All Six Models	**Average Single Model**
Random	00:05:50	00:00:58
BADGE	00:07:52	00:01:19
CAL	00:14:59	00:02:30
ALPS	00:07:21	00:01:14
Active PETs	00:08:01	00:01:20
Active PETs-o	00:09:10	00:01:32

Table D.3: Average run time for a single iteration for each of the sampling methods. The time format is hours:minutes:seconds.

	CFEVER	**SciFact_retrieved**	**SciFact_oracle**
Active PETs-o	05:53:08	04:12:33	02:31:27

Table D.4: Total run time for running Active PETs with oversampling iteratively up to 300 instances on different datasets. The time format is hours:minutes:seconds.