UNIVERSITÉ
LIBRE
DE BRUXELLES

BRUFACE
BRUSSELS FACULTY
OF ENGINEERING

VRIJE
UNIVERSITEIT
BRUSSEL

Graduation thesis submitted in partial fulfilment of the requirements for the degree of engineering sciences: Electrical Engineering

Towards neural architectures for Fact Extraction and VERification

Scientific Fact Verification

Boya Zhang

Master thesis submitted under the supervision of
Prof. Dr. Ir. Nikolaos Deligiannis

The co-supervision of
Dr. Ir. Giannis Bekoulis

Academic year
2020

In order to be awarded the Master's programme in
Electrical Engineering

Declaration

‘This master’s thesis came about (in part) during the period in which higher educa- tion was subjected to a lockdown and protective measures to prevent the spread of the COVID-19 virus. The process of formatting, data collection, the research method and/or other scientific work the thesis involved could therefore not always be carried out in the usual manner. The reader should bear this context in mind when reading this Master’s thesis, and also in the event that some conclusions are taken on board’.

Abstract

In order to perform automatic fact checking on scientific claims, we study on the Fact Extraction and VERification task with neural architecture implementation. Tasks with corpus including FEVER and SCIFACT have given a baseline approach for conducting the experiments. We first select the relevant abstracts or documents from the corpus, then retrieve evidence sentences from each selected abstracts or documents. In the end, we verify the claim with its evidence sentences and label the claim into three classes: SUPPORTS, REFUTES and NOT ENOUGH INFO. FEVER task contains a large scale dataset extracted from the Wikipedia database. The study on document retrieval has already achieved fair enough accuracy with mention or keyword based methods. The work on evidence sentences selection and claim verification is mainly conducted with ESIM and language model based methods. The research on FEVER task offers groundwork for scientific fact extraction. And SCIFACT provides scientific claim verification task generated from credible article sources. The baseline model VERSCI and improved model SciKGAT have given state of art accuracy for the task. They both employ the methods that have been proved to be useful on the FEVER task. We focus on the label prediction step for SCIFACT task with the Transformer-XH model. We first follow the work of VERSCI to retrieve the evidence sentences with TF-IDF and BERT-style model. We then construct the evidence graph with golden and retrieved evidence sentences. And we later concatenate the SCIFACT and FEVER evidence graph to train the Transformer-XH model for better recognition of textual entailment connection on scientific content. Transformer-XH inherits from Transformer and BERT with extra hop attention as an innovation. This helps to identify relationships among different evidence sentences. Therefore it enhances the accuracy of multi evidence reasoning. Our trained model gives the state of art accuracy 78.0 on the full claims and provides better accuracy 82.35 on the multi evidence claims.

Keywords: Scientific claim verification, Transformer, BERT, Attention mechanism

Acknowledgements

I must express my very big thanks to supervisor Dr. Ir. Giannis Bekoulis and promoter Prof. Dr. Ir. Nikolaos Deligiannis. They are very patient and are always there to help when I need instructions or equipment. Dr. Giannis Bekoulis always show me the direction for the research when I feel overwhelmed and answer to my questions in details. In October and November, we set up meetings almost every week to perform the thesis step by step. In December, even though on his vacation, Giannis always write back to my emails quickly and follow up with my work. Prof. Dr. Ir. Nikolaos Deligiannis teaches my master courses Digital Singal Processing and Machine Learning and Big Data Processing and also instructed me with the Project Electronics and Telecommunication in master year 1. I have to say that at first it took me time to adapt to the study and life in Belgium. But Nikos is always very patient and kind to answer questions, and I improved a lot with my study outcomes under his guidance.

Thank you my friends and family for cheering me up when I met difficulty and always by my side to listen when I want to express my feelings.

Grateful for the opportunity to study at VUB and thanks to NPU for offering me the opportunity for an exchange program in the first place.

Thank you my mom.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Fact Extraction and VERification is a Natural Language Processing task to extract evidence from reliable corpora in order to identify the facticity of existing claims. This task is useful because there exist a lot of misleading information and even intentionally created fake news and arguments on the internet. Not everyone has the knowledge to discern the truth and sometimes the misleading information in the virtual world could lead people in the real world to tragedy incidents. Thus we need to find a way to identify the possible misleading arguments on the internet and stick a label on to them, or create a system for people to search for the facticity of a certain claim and give them the evidence behind the scene.

Since the amount of misleading information is with huge number, it would be quantitatively impossible to hire people perform checking and stick labels all the time, meanwhile the quality of manual review can be high and low. Who can perform fact checking for such huge number of information and meanwhile maintain a relatively stable quality? Fact Extraction and VERification performed by computer is the answer.

1.1 Problem Statement

In this paper we study Fact Extraction and VERification based on the in-domain datasets consist of claims accompanied by three types of label, evidence sentences and reliable documents or abstracts. Our task is to achieve better results on the evaluation process with the models that we use. The dataset we are exploring is the SCIFACT dataset with the scientific specific terms in the content. An example of scientific claim verification is illustrated in Figure 1.1. The FEVER dataset is introduced for the advanced methods that have been deployed on the FEVER task and the advanced outcomes have been obtained. The study on FEVER dataset brings a certain significance to our study on SCIFACT dataset. And when we train the FEVER dataset together with the SCIFACT dataset, it extended the ability to acknowledge the textual entailment relation between claims and evidence sentences.

1.2 Objective and Contribution

The goal of this paper is to classify the SCIFACT claims into three classes with three steps of procedure. The three classes are SUPPORTS, REFUTES and NOT ENOUGH INFO. The three

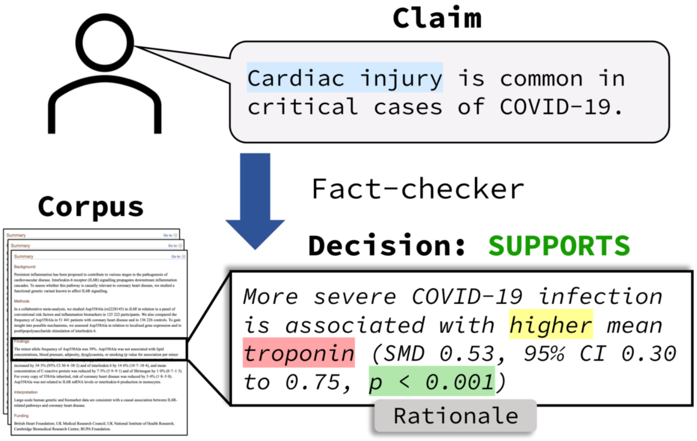

Claim
Cardiac injury is common in critical cases of COVID-19.
Fact-checker
Corpus
Decision: SUPPORTS
More severe COVID-19 infection is associated with higher mean troponin (SMD 0.53, 95% CI 0.30 to 0.75, p < 0.001)
Rationale

Figure 1.1: An example of scientific claim verification (Wadden et al., 2020).

steps are abstract retrieval, rationale selection and label prediction.

We exploited the first two steps with the baseline model VERSCI. We then focus on the work of label prediction on SCIFACT task with the model Transformer-XH. Since the model has achieved state of art performance on the third step of FEVER task and meets the particularities of SCIFACT dataset.

1.3 Thesis Structure

We formulate the paper with five chapters.

Chapter 1 - Introduction In the Introduction Chapter, we first show the definition of Fact Extraction and VERification. We then explain the reason of studying this field of research and its benefit to the human society. Next, we state the specific research problem on scientific dataset and neural architecture models. Later, we give a brief preface to our methods employed. In the end, the structure of the paper is introduced.

Chapter 2 - Related Work In the Related Work Chapter, we bring in the FEVER and SCIFACT task and existing models on them. We then discussed about the similarities and differences between the two tasks and make a conclusion regarding the choice of neural architectures on the SCIFACT task.

Chapter 3 - Models In the Models Chapter, we show the structure of preliminary neural architectures: Transformer and BERT. They are the basis of the models that we carried out on the Abstract Retrieval, Rationale Selection and Label Prediction step. Then we discuss about the TF-IDF, SCIBERT and RoBERTa-large method for retrieving the evidence sentences. In the end with the claim and the evidence sentences we constructed the evidence graphs and implement the Transformer-XH on label prediction.

Chapter 4 - Results and Discussion In the Results and Discussion Chapter, we illustrate the experimental setups including dataset details and evaluation methods. Then we introduce the implementation process with the training procedure and the parameters applied. In the end, the results of the experiments are depicted and evaluated.

Chapter 5 - Conclusion In the Conclusion Chapter, we give a review on the existing progress and the future work.

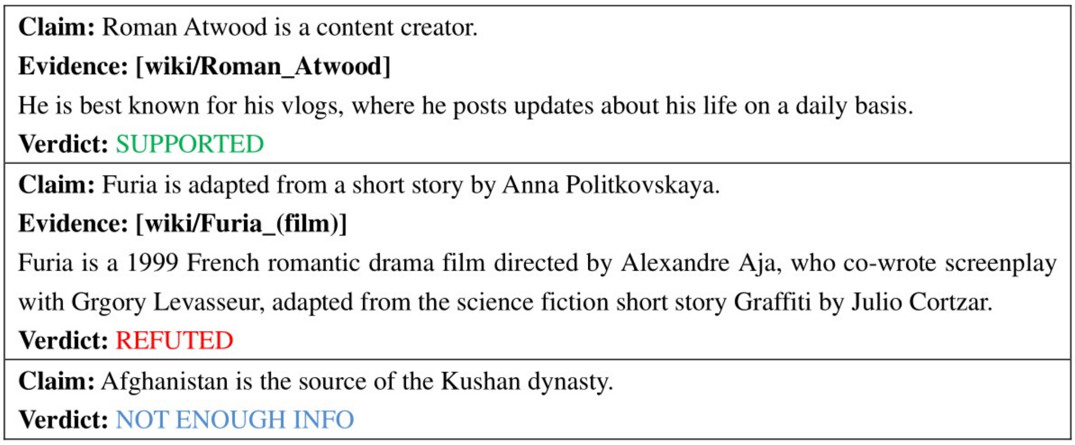
Chapter 2

Related Work

In this chapter, we first discuss about a large-scale dataset called FEVER coming from the reliable source, Wikipedia. The dataset consist of labeled claims with golden evidences. We then bring in the FEVER task with three steps: documents retrieval, evidence sentences selection and claim verification. Secondly, we give an overview on existing methods and models regarding divided FEVER task steps. Most of the models for document retrieval are either mention-based or keyword-based. For sentence-level evidence selection and claim verification, the models are mostly ESIM-based or Language Model based. But there are also joint setting models. We then discuss about the result for each step and the overall pipeline. In the end, we bring in the scientific dataset SCIFACT coming from reliable sources of scientific articles. This dataset is similar to the FEVER dataset from the perspective of dataset component, pipeline structure and predicted label. But it contains more specific scientific terms thus is more complicated for prediction. There are not yet many models implemented on the dataset. We then introduced the three steps for SCIFACT task: abstract retrieval, rationale selection and label prediction. Next, we give an overview on existing methods and models regarding SCIFACT task. As there are not much models specifically trained on the dataset, we depicted two existing models: VERSCI and SciKGAT. In the end, we close this chapter with the discussion of similarities and differences between FEVER and SCIFACT task, and what can be learnt from the existing developed methods on FEVER to SCIFACT.

2.1 FEVER: A Large-Scale Dataset for Claim Verification on Textual Sources

FEVER is a large-scale English dataset contains 185445 claims generated by human annotators with sentences extracted from Wikipedia, the free encyclopedia (Thorne et al., 2018). The claims are verified, without knowing the sentences where the claims were derived from, by separate annotators into classes as SUPPORTED, REFUTED or NOT ENOUGH INFO, and with sentences recorded as the evidence for the SUPPORTED and REFUTED classification. Three examples from FEVER dataset are depicted in Figure 2.1.



Claim: Roman Atwood is a content creator.

Evidence: [wiki/Roman_Atwood]

He is best known for his vlogs, where he posts updates about his life on a daily basis.

Verdict: SUPPORTED

Claim: Furia is adapted from a short story by Anna Politkovskaya.

Evidence: [wiki/Furia_(film)]

Furia is a 1999 French romantic drama film directed by Alexandre Aja, who co-wrote screenplay with Grgory Levasseur, adapted from the science fiction short story Graffiti by Julio Cortzar.

Verdict: REFUTED

Claim: Afghanistan is the source of the Kushan dynasty.

Verdict: NOT ENOUGH INFO

Figure 2.1: Different types of claims in FEVER dataset. For each claim, the verdict is recorded. For the claims with SUPPORTED or REFUTED verdict, the name of the document(s) and the evidence sentence(s) are recorded.

2.1.1 Dataset Construction

The construction of FEVER dataset consists of manually Claim Generation and manually Claim Labeling.

- Claim Generation: Extracting information from Wikipedia and generating claims from it (Thorne et al., 2018).

- Claim Labeling: Classifying whether a claim is supported or refuted by Wikipedia and selecting the evidence for it, or deciding there's not enough information to make a decision (Thorne et al., 2018).

For the reliability of the constructed dataset, as the human annotators are trained or experienced native US English speakers and the data validation process contains 5-way inter-annotator agreement, agreement against super-annotators and manual validation by the authors. Also the Fleiss κ score for the 5-way agreement is 0.6841, which is convincing given the complexity of the task (Fleiss, 1971).

The constructed dataset is then divided into reserved set, training set, development set[1] and test set. The three useful sets for our task are shown in Table 2.1. The golden document(s), golden evidence sentence(s) and golden label of the training and development set are provided publicly for researchers. The ones of test set are kept secret in order to evaluate the result handed in for the FEVER Challenge.[2] As for the reserved set, it is for another shared task.

[1] It is also called validation set.

[2] https://competitions.codalab.org/competitions/18814

Set	SUPPORTED	REFUTED	NOT ENOUGH INFO	Total
Training	80,035	29,775	35,639	145,449
Dev	3,333	3,333	3,333	9,999
Test	3,333	3,333	3,333	9,999
Total	86,701	36,441	42,305	165,447

Table 2.1: Training, Dev and Test Set Sizes for Each Classes of the FEVER Dataset (Thorne et al., 2018).

2.1.2 Baseline System

The baseline system of FEVER dataset contains three components: document retrieval, sentence-level evidence selection[3] and claim verification[4] as depicted in Figure 2.2. **The document retrieval step** aims at taking claim and Wikipedia documents as input, matching the query with the collection of unstructured documents and returns the most relevant documents as output. **The sentence-level evidence selection step** aims at taking claim and the selected documents as input, matching the query with sentences and returns the most relevant sentences. **The claim verification step** aims at taking claim and retrieved sentences as input, identify the veracity of the claim and return the predicted label.

2.1.3 Baseline Model and Evaluation

Baseline Model

The baseline model of FEVER is provided along with the dataset and the baseline system. For **document retrieval**, the DrQA module is implemented to obtain the k most similar documents, and it use cosine similarity based on the TF-IDF word representation (D. Chen et al., 2017). For **sentence-level evidence selection**, the unigram TF-IDF vectors or bigram TF-IDF with binning are implemented to obtain the l most similar sentences. For **claim verification**, the multi-layer perception (MLP) model (Riedel et al., 2017) or the decomposable attention (DA) model (Parikh et al., 2016) are implemented to obtain the classification label.

Evaluation Method

For the evaluation process[5] of the FEVER task, the performance in each sub-task and also the full pipeline are evaluated. The performance of the previous step will have positive influence on the next step. For instance, if the performance in the document retrieval is better, it will lead to better performance in the sentence retrieval and claim verification.

For document retrieval evaluation, the *fully supported* and *oracle accuracy* are introduced.*Fully supported* only considers the claims with SUPPORTED and REFUTED label, and the accuracy reflects whether the document(s) is(are) fully correctly retrieved. *Oracle accuracy* considers all the three classes and define that the document retrieved for NOT ENOUGH INFO class are

[3] It is also called rationale selection.

[4] It is also called textual entailment or label prediction.

[5] The code of the evaluation module is available at: https://github.com/sheffieldnlp/fever-scorer

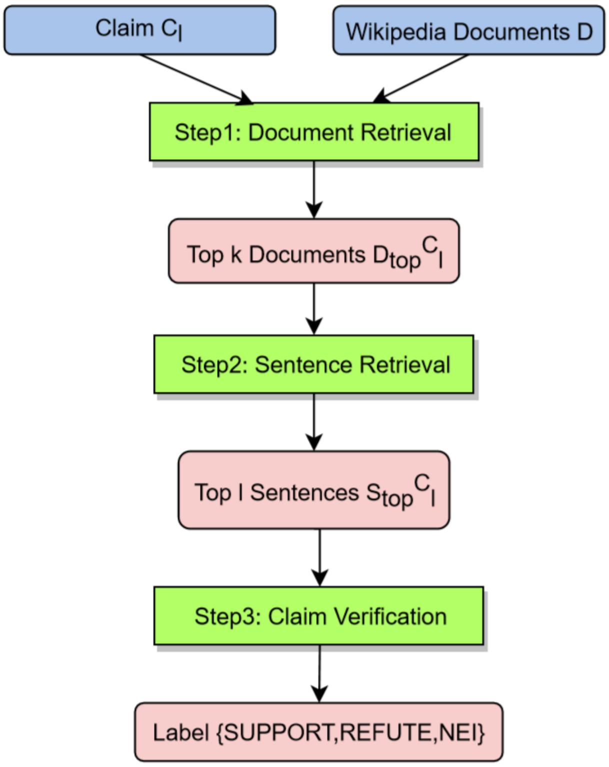

Claim C_l
Wikipedia Documents D
Step1: Document Retrieval
Top k Documents $D_{top}^{C_l}$
Step2: Sentence Retrieval
Top l Sentences $S_{top}^{C_l}$
Step3: Claim Verification
Label {SUPPORT,REFUTE,NEI}

Figure 2.2: The baseline system of FEVER dataset.

always correct, thus it is the upper bound accuracy.

For sentence-level evidence selection, the *precision*,*recall* and F_1 scores are introduced. The *precision* is the percentage of correctly retrieved sentences over all the retrieved sentences for claims with SUPPORTED and REFUTED label. It represents the probability of relevant retrieval. The *recall* is the correctly retrieved sentences over all the evidence sentences(whether they are retrieved or not). It represents the probability of complete retrieval. The F_1 scores approximately the average of *precision* and *recall*.

For claim verification, the *label accuracy* and *FEVER score metrics* are introduced. The *label accuracy* simply measures the accuracy of the label prediction. The *FEVER score metrics* considers the claim to be correct only if a complete evidence group has been correctly retrieved and the label is correct. The *FEVER score metrics* is a higher standard compared to *label accuracy*.

For the full pipeline, the FEVER score is the primary metric for the evaluating and ranking of proposed system.

Baseline Model Evaluation

The evaluation on the baseline model is performed.

- Whether the system is trainable: With fewer than 6000 training instances, the accuracy of DA or MLP is unstable. But as the train instances increase to a larger number, the accuracy increases with the logarithm of the number of train instances.

- Whether the system is challenging: The best *label accuracy* for the baseline model is 50.91%. The best *FEVER score* for the baseline model is 31.87%.

After evaluating the system on baseline model through pipeline approach, the task is proved to be difficult but still feasible. So it can help develop models for fact extraction and verification meanwhile distinguish the different learning capabilities of different models.

2.2 Existing Models on FEVER

Most of the existing methods that have been developed for FEVER task divide the task into three steps (document retrieval, sentence-level evidence selection and claim verification) same as the baseline system. But there are also studies that merge the sentence-level evidence selection and claim verification together with multi-task learning architectures (Bekoulis et al., 2020).

2.2.1 Models for Document Retrieval

For the document retrieval step, the main methods are mention-based, keyword-based and others (Bekoulis et al., 2020).

Mention-Based

The mention-based approach proposed by Hanselowski et al., 2018 have three components.

- Mention extraction: The noun phrases in the claim are all considered as candidates for potential entities.

- Candidate article search: An external search API is used for matching the potential entity in the Wikipedia titles.

- Candidate filtering: All titles which are not part of the claim are discarded.

The value of name entities and disambiguation information are exploited by Malon, 2018 and Chakrabarty et al., 2018.

Keyword-Based

Nie et al., 2018a exploited with exact matching, article elimination and singularization to perform keyword-matching. Luken et al., 2018 extracted part-of-speech tags and dependencies with CoreNLP parser (Manning et al., 2014).

Others

Other methods include:

- *extra matching techniques* designed for higher precision (Taniguchi et al., 2018)
- *hand-crafted features* such as position and capitalization in the claim, which then trained on neural methods (Hidey and Diab, 2018) or logistic regression classifier (Yoneda et al., 2018).

2.2.2 Models for Sentence-Level Evidence Selection

For the sentence-level evidence selection, the main methods are TF-IDF, ESIM-based, Language Model Based and others (Bekoulis et al., 2020).

TF-IDF

Many methods use a cosine similarity function together with a TF-IDF vector (Taniguchi et al., 2018; Yin and Schütze, 2018; Portelli et al., 2020). Also ELMo embedding is exploited by Chakrabarty et al., 2018.

ESIM-Based

Enhanced LSTM for Natural Language Inference (ESIM) models transform the sentence level evidence selection step into a Natural Language Interface (NLI) problem with claim as ”premise” and candidate evidence sentences as ”hypothesis”(Q. Chen et al., 2017).

The modified version of ESIM made some modifications on the input part of candidate evidence sentences(Hanselowski et al., 2018). The candidate evidence sentences include the positive samples and the negative samples. The positive samples are the ground truth evidence sentences. The negative samples are 5 randomly selected sentences from the Wikipedia document where the positive samples are in. The positive samples are not included for the selection of negative samples. The loss function used is a pairwise hinge loss that take the positive and negative ESIM ranking scores as input. The testing computes the ranking score between the claim and each candidate evidence sentence.

Neural Semantic Matching Networks (NSMN) is a variation of ESIM. First, the NSMN score is calculated between the claim and the evidence sentences. Second, the highest scoring sentences are retained with a threshold-based prediction. A cross-entropy loss is exploited when training the model (Nie et al., 2018b).

Language Model Based

The language model based methods also transform the sentence level evidence selection step into a Natural Language Interface (NLI) problem. The pre-trained language models including BERT (Devlin et al., 2018), RoBERTa (Y. Liu et al., 2019) and XLNet (Yang et al., 2019) are finetuned for the task.

On one hand, the pointwise loss is used not only on the BERT model but also on the RoBERTa and XLNet model: if the claim and the candidate evidence sentence are related, a cross-entropy classifier would predict 1, otherwise a 0 would be predicted.

On the other hand, the pairwise loss is used only on the BERT model: The loss function takes in a pair of positive and negative example. The positive example is the concatenation of the claim and the positive sample. The negative example is the concatenation of the claim and the negative sample. The goal is to make the model learn to maximize the margin between the positive and negative examples. In the work of Soleimani et al., 2019, they select more difficult negative examples with highest loss value, similar to hard negative mining (Schroff et al., 2015).

Others

Other methods include:

- the two-step model combining the ESIM-based and the LM-based sentence retriecal components of Stammbach and Neumann, 2019
- the logistic regression model fed with different features.

2.2.3 Models for Claim Verification

For the claim verification, the main methods are ESIM-based, Language Model based and other neural models (Bekoulis et al., 2020).

ESIM-Based

The ESIM model used by Hanselowski et al., 2018 including attention mechanism, pooling operations and an MLP classifier. Also, the NSMN used by Nie, Chen, et al., 2019 exploits additional features like WordNet embeddings, number embeddings and the scores from the previous steps. In addition, the ESIM model used by Yoneda et al., 2018 consider each candidate evidence to be independent from the claim and use an MLP classifier in addition to the prediction score of each evidence sentence.

Language Model Based

Following models for the step are all BERT-based models. And almost all the language models used in the claim verification step are BERT-based models.

- Soleimani et al., 2020 takes claim verification as an NLI task with claim as premise and candidate evidence sentences as hypothesis. The evidences are considered to be independent from the claim and the label classification follows an aggregation rule.
- Zhou et al., 2019 use GNNs so that evidences become nodes in a graph, then the information between nodes can be exchanged.
- Z. Liu, Xiong, Sun, and Liu, 2020 use Kernal attention at sentence and token level, and propagate information among the evidence nodes.

- Zhong et al., 2019 construct the graph with semantic roles and then the GNNs and graph attention methods are used to aggregate the information from graph nodes.
- Zhao et al., 2020 use extra hop Transformers to perform multi-hop reasoning on text sequences, thus information from different sentences and documents is combined.

Other Neural Models

Other neural models include DA model, bidirectional LSTMs, Convolution Neural Networks (CNNs) in combination with attention methods and Transformers.

2.2.4 Joint Setting Models

The FEVER subtasks are also handled by some teams in a joint setting because there are errors flowing from one step to the other when implementing the standard pipeline approach (Bekoulis et al., 2020).

2.2.5 Existing Results Illustration

Document Retrieval

The existing best Fully Accuracy score 93.55 comes from the mention-based method brought up by Hanselowski et al., 2018. And the best Oracle Accuracy score 94.40 comes from the work of Chakrabarty et al., 2018. We can conclude that the document retrieval step have already obtained quite good performance.

Standard	Fully Accuracy	Oracle Accuracy
Best score	93.55	94.40

Table 2.2: Best Scores for Document Retrieval

Sentence-Level Evidence Selection

The sentence-level evidence selection result evaluation are performed both on the dev set and the test set. With the same method, the scores on the dev set are higher than the scores on the test set. The best Precision is 77.50 on dev set and 77.23 on test set. The work that gives this result is conducted by Luken et al., 2018. The best Recall is 94.37 on dev set and 87.47 on test set. The result comes from the work of Z. Liu, Xiong, Sun, and Liu, 2020. And in the end the best F_1 Score is 76.87 on dev set and 74.62 on test set. The work is from Nie, Wang, et al., 2019. We can conclude that the sentence selection step have already obtained quite good performance but there is still a little bit space for improvement.

Dataset	Dev			Test		
Standard	Precision	Recall	F_1	Precision	Recall	F_1
Best score	77.50	94.37	76.87	77.23	87.47	74.62

Table 2.3: Best Scores for Sentence-Level Evidence Selection

Claim Verification

The claim verification result evaluation are performed both on the dev set and the test set. For the same method, the scores on the dev set are higher than the scores on the test set. The best Label Accuracy is 84.33 from Portelli et al., 2020. They didn't release their result for the FEVER score on dev set nor the FEVER score and Label Accuracy on the test set, otherwise the best results for the rest of the comparison could also come from their work. The best FEVER score on dev set is 76.11 from Z. Liu, Xiong, Sun, and Liu, 2020. The best Label Accuracy on test set is 76.85 from Zhong et al., 2020. And the best FEVER score on test set is 74.27 from Stammbach and Ash, 2020. We can conclude that we can work on making enhancement on the result of claim verification step.

Dataset	Dev		Test	
Standard	Label Accuracy	FEVER	Label Accuracy	FEVER
Best score	84.33	76.11	76.85	74.27

Table 2.4: Best Scores for Claim Verification

2.2.6 Conclusion

All the aforementioned methods can not only be applied on the FEVER task but also make contribution to the FEVER related tasks from different aspects of automated fact checking.

2.3 SCIFACT: A Dataset Designed for Verification on Scientific Claims

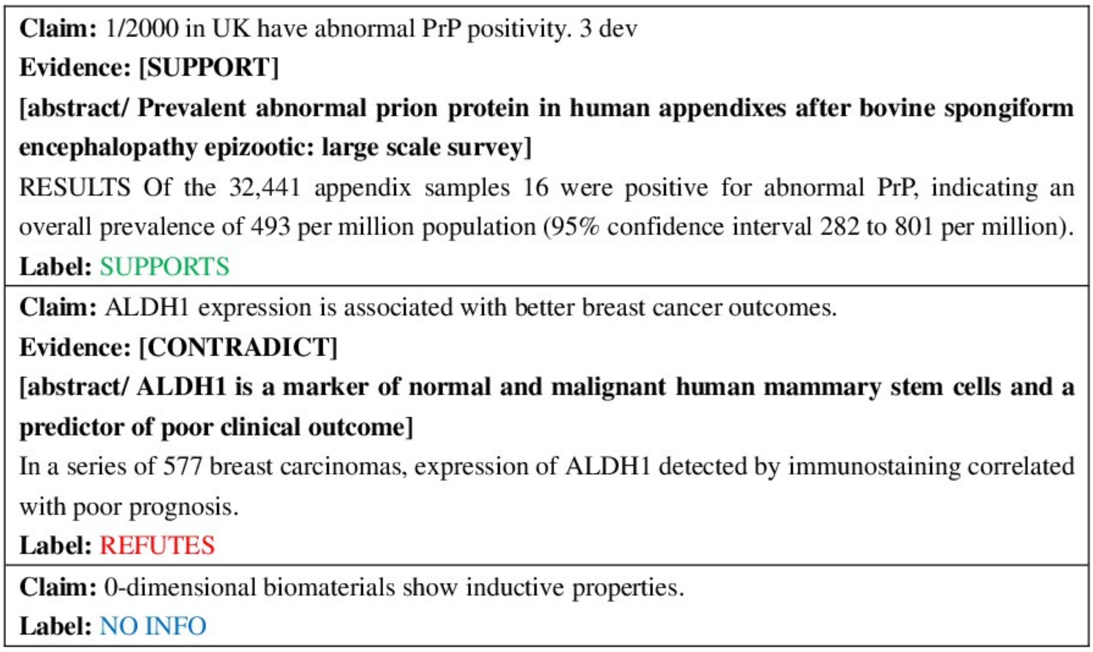
SCIFACT is a dataset contains 1,409 scientific claims verified through a corpus of 5,183 abstracts. The claims are accompanied with abstracts and rationales that either SUPPORTS or REFUTES the claim. For the claims with NO INFO, no abstract or rationale is provided. The examples for SciFact dataset is depicted in Figure 2.3. For each claim, the evidence sentence(s) and the abstract(s) are present together with [SUPPORT] or [CONTRADICT] to the claim. The label SUPPORTS, REFUTES or NO INFO is interpreted from the evidence(s). Although with this structure a claim can be supported or refuted by different abstracts and rationales, but in the SciFact dataset a claim is always only supported or refuted by its abstracts and rationales.

2.3.1 Dataset Construction

The claims are generated by expert human annotators from citation sentences of scientific literature. The construction of SCIFACT dataset contains **corpus creation** and **annotation** process.

Corpus Creation

The corpus consist of *seed set abstracts*, *co-cited article abstracts* and *distractor abstracts*. The publicly-available large-scale corpus S2ORC (Lo et al., 2020) is used to construct the SciFact dataset. The corpus creation process is shown in Figure 2.4.



Claim: 1/2000 in UK have abnormal PrP positivity. 3 dev

Evidence: [SUPPORT]

[abstract/ Prevalent abnormal prion protein in human appendixes after bovine spongiform encephalopathy epizootic: large scale survey]

RESULTS Of the 32,441 appendix samples 16 were positive for abnormal PrP, indicating an overall prevalence of 493 per million population (95% confidence interval 282 to 801 per million).

Label: SUPPORTS

Claim: ALDH1 expression is associated with better breast cancer outcomes.

Evidence: [CONTRADICT]

[abstract/ ALDH1 is a marker of normal and malignant human mammary stem cells and a predictor of poor clinical outcome]

In a series of 577 breast carcinomas, expression of ALDH1 detected by immunostaining correlated with poor prognosis.

Label: REFUTES

Claim: 0-dimensional biomaterials show inductive properties.

Label: NO INFO

Figure 2.3: Three Examples from SCIFACT Dataset.

The first process is article selection, the articles with at least 10 citations are randomly selected from a manually organized compilation of reliable journals(Wadden et al., 2020). The selected articles are called *seed set*.

Secondly, the S2ORC citation graph is used to sample *source citations* in the *citing articles* which cite the *seed set*. The claim is generated from the *source citation* and are natural as the *source citation* take place naturally in a scientific article.

Thirdly, other articles the *source citation* cites but not in the *seed set* are described as *co-cited articles*. In most of the *source citations*, the *seed set* is only used and it tends to create claims that are specific and more verifiable.

Lastly, the corpus are further expanded with *distractor abstracts* by identify 5 papers cited in a different paragraph from *source citation* in the same *citing article*.

Annotation Process

The annotation process contains claim writing, claim negation, claim verification and quality assessment.

Claim writing: The scientific claim is defined as an *atomic verifiable statement* phrasing one point of a scientific individual or procedure, which can be validated with a particular source

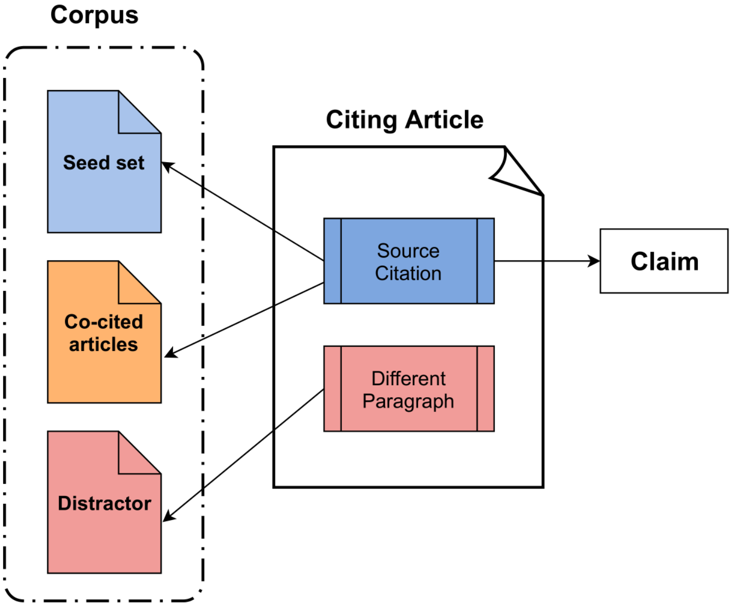


Figure 2.4: Corpus Creation.

(Wadden et al., 2020). The annotators professional in scientific NLP and life sciences write the scientific claims from the *source citations.*

Claim negation: The claims written from *source citations* are always supported or cannot be verified by the *seed set abstracts* and *co-cited article abstracts* since the scientific articles are reliable. In order to obtain claims with REFUTES verdict, the annotators wrote the negation claims from the generated scientific claims.

Claim verification: The annotators found evidence in 63% of cited abstracts and each rationale sentence is recorded for either SUPPORT or CONTRADICT towards the claim. Although with this structure a claim can be supported or refuted by different abstracts and rationales, but in the SciFact dataset a claim is always supported or refuted by its abstracts and rationales. Claims which rationales cannot be found are only accompanied with cited abstracts but not rationales.

Quality: For abstract agreement, the label agreement is 0.75 Cohen's κ (Wadden et al., 2020). For rationale agreement, the sentence-level agreement is 0.71 Cohen's κ (Wadden et al., 2020). Therefore, the dataset is convincing.

Dataset

The constructed dataset is then divided into training set, development set and test set as depicted in Table 2.5. The golden abstract(s), golden evidence rationale sentence(s) and golden label of the training and development set are provided publicly for researchers. The ones of test set are kept secret in order to evaluate the result handed in for the SciFact Challenge[6].

[6] https://scifact.apps.allenai.org/leaderboard

Set	SUPPORTS	REFUTES	NO INFO	Total
Training	332	304	173	809
Dev	124	112	64	300
Test	100	100	100	300
Total	556	516	337	1409

Table 2.5: Training, Dev and Test Set Sizes for Each Classes of SciFact Dataset.

The evidence granularity for SCIFACT dataset is illustrated below in an intuitive way. We can infer that most of the claims are related to only one cited abstract as shown in Table 2.6, and most of the claims with evidence accompanied have only one evidence abstract as shown in Table 2.7. We can observe that most of the abstracts are distractors or contain no rationales and among evidence abstracts, most abstracts contain only one rationale as shown in Table 2.8. We can also conclude that most rationales contain only one sentence as shown in Table 2.9.

Cited Abstracts per Claim	0	1	2	3+	Total
Number of Claims	-	1278	86	45	1409

Table 2.6: Claim Sizes for Each Classes of Cited Abstracts per Claim.

Evidence Abstracts per Claim	0	1	2	3+	Total
Number of Claims	516	830	37	26	1409

Table 2.7: Claim Sizes for Each Classes of Evidence Abstracts per Claim.

Rationales per Abstract	0	1	2	3+	Total
Number of Abstracts	4188	552	290	153	5183

Table 2.8: Abstract Sizes for Each Classes of Rationales per Abstract.

Sentences per Rationale	0	1	2	3+	Total
Number of Rationale	-	1542	92	11	1645

Table 2.9: Rationale Sizes for Each Classes of Sentences per Rationale.

2.3.2 Baseline System

The baseline system of SCIFACT dataset contains three components: abstract retrieval, rationale selection and label prediction as shown in Figure 2.5.

The document retrieval step aims at taking the claim and all the abstracts as input, matching the query with the collection of corpus and returns the most relevant abstracts as output.

The rationale selection step aims at taking claim and the selected abstracts as input, matching the query with sentences and returns the most relevant rationale sentences. A rationale R is a collection of sentences S.

The label prediction step aims at taking claim and retrieved rationale sentences as input, identify the veracity of the claim and return the predicted label.

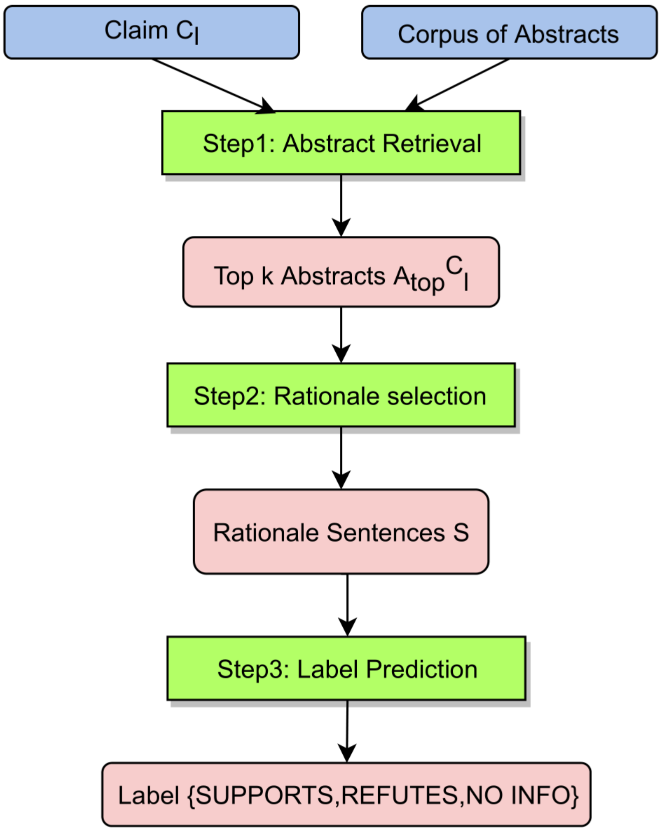

Claim C_I
Corpus of Abstracts
Step1: Abstract Retrieval
Top k Abstracts $A_{top}{}^{C_I}$
Step2: Rationale selection
Rationale Sentences S
Step3: Label Prediction
Label {SUPPORTS,REFUTES,NO INFO}

Figure 2.5: The Baseline System of SciFact Dataset.

2.3.3 Evaluation Method

The SCIFACT task is evaluated with two levels of granularity: the *abstract-level* which is the "Open" FEVER style (Thorne et al., 2018) and the *sentence-level* which is the "Oracle abstract" ERASER-style (DeYoung et al., 2020).

Abstract-Level

For the claim c, the predicted evidence abstract a is *correctly-labeled* when these conditions are met:

- a is a golden abstract of c,
- prediction of the label is accurate.

In addition, after meeting the above requirements, if the rationales in the abstracts are retrieved correctly, the abstracts are then *correctly-rationalized.* The maximum number of predicted rationale sentences is three.

Overall performance is estimated with precision, recall and micro-F1 score. The correctly labeled abstracts are referred to as $Abstract_{Label-Only}$, and the correctly labeled and rationalized abstracts are referred to as $Abstract_{Label+Rationale}$.

Sentence-Level

For the claim c, the predicted evidence sentence s is *correctly-selected* when these conditions are met:

- s is a golden evidence sentence of a golden rationale R,
- other evidence sentences in the golden rationale R are retrieved,
- this not include the NO INFO condition.

In addition, after meeting the above requirements, if the label is predicted correctly, the predicted evidence sentences are then *correctly-labeled.* The maximum number of predicted rationale sentences is not limited but the evaluation penalizes the over-predicted rationale sentences.

Overall performance is studied with precision, recall and micro-F1 score. The correctly-selected sentences are referred to as $Sentence_{Selection-Only}$, and the correctly selected and labeled sentences are referred to as $Sentence_{Selection+Label}$.

2.4 Existing Models on SCIFACT

Existing models on SCIFACT are VERSCI (Wadden et al., 2020) and SciKGAT (Z. Liu, Xiong, Dai, et al., 2020). The VERSCI is the baseline model proposed together with the dataset. SciKGAT implemented the state-of-art model KGAT (Z. Liu et al., 2019) on SCIFACT dataset with improvement of more than 10% absolute F1 score and 30% absolute precision. On VERSCI the precision is 46.6%, SciKGAT improved it to 76%.

2.4.1 VERSCI

VERSCI is the baseline model of the SCIFACT dataset. It follows the baseline system structured with 3 steps: abstract retrieval, rationale selection and label prediction. The model used for the abstract retrieval is TF-IDF, for the rationale selection and label prediction are separate BERT language models.

Abstract Retrieval

For the abstract retrieval, k abstracts are selected base on the TF-IDF resemblance with the claim.

Rationale Selection

For the rationale selection, we need to take claim and the selected abstracts as input, match the query with sentences and return the most relevant rationale sentences. Thus the goal is to predict score $z_i = 1$. A BERT-style language model is implemented to encode the concatenation between sentence a_i and the claim c, where w_i is the encoded sequence (Wadden et al., 2020).

$$w_i = [a_i, SEP, c] \tag{2.1}$$

Score $\hat{z_i}$ is predicted with Equation 2.2, where σ is the sigmoid function, f is a linear layer and $CLS(w_i)$ is the CLS token from the encoded sequence w_i.

$$\hat{z_i} = \sigma[f(CLS(w_i))] \tag{2.2}$$

The model is trained on claims together with their cited abstracts. The cross-entropy loss of $\hat{z_i}$ and z_i is minimized. For each claim, the negative examples are from non-rationale sentences for the claim. The sentences a_i with $\hat{z_i}>t$ is selected for evidence sentence $\hat{s_l}$, where $t \in [0, 1]$ is the threshold.

Label Prediction

For the label prediction, we need to take claim and retrieved rationale sentences as input and return the predicted label. Thus the goal is to predict true Label y. A separate BERT-based model is implemented to encode the concatenated sequence of the sentence s_l and the claim c, where u is the encoded sequence.

$$u = [\hat{s_1}, ..., \hat{s_l}, SEP, c] \tag{2.3}$$

Label $\hat{y}$ is predicted with Equation 2.4, we have ϕ as the softmax function, f as one linear layer containing three outputs being three different labels and $CLS(u)$ is the CLS token from the encoded sequence u.

$$\hat{y} = \phi[f(CLS(u))] \tag{2.4}$$

The model is trained on claims together with their evidence sentences. The cross-entropy loss of $\hat{y}$ with y is minimized. For claims with rationales, the evidence sentences are golden evidence sentences for the claim. For NO INFO claims, the evidence sentences are k sentences obtained highest TF-IDF similarity regarding the claim. The predicted label is y_p = argmax$\hat{y}$.

2.4.2 SciKGAT

Abstract Retrieval

For the abstract retrieval, the inputs are claim c and abstract $D = a_1, ..., a_l$, the aiming outputs are three most relevant evidence abstracts.

The first step is to retrieve 100 most relevant abstracts with TF-IDF from D. Then the claim c and abstract a with title t are concatenated and encoded with BERT model, where H is the encoded sequence.

$$H = BERT([CLS], c, [SEP], t, a, [SEP]) \tag{2.5}$$

The encoded sequence contains tokens from both the claim and abstract. The relevance label y_a between claim c and abstract a is calculated, where H_0 is the [CLS] token in H.

$$p(y_a|c, a) = softmax_{y_a}(MLP(H_0))) \tag{2.6}$$

The abstracts are reranked from high to low with probability $p(y_a = 1|c, a)$ and we have the top-3 abstracts.

Rationale Selection

For the rationale selection, the inputs are the 3 retrieved abstracts $a = e_1, ..., e_k$ and the claim c, the aiming output is the evidence set $E = e_1, ..., e_q$ for each abstract a.

The claim c and evidence e are concatenated and encoded with BERT model, where H is the encoded sequence.

$$H = BERT([CLS], c, [SEP], e, [SEP]) \tag{2.7}$$

The encoded sequence contains tokens from both the claim and evidence. The relevance label y_r between claim c and evidence e is calculated, where H_0 is the [CLS] token in H.

$$p(y_r|c, e) = softmax_{y_r}(MLP(H_0))) \tag{2.8}$$

The evidences with $(p(y_r = 0|c, e) < p(y_r = 1|c, e))$ are retrieved and forming the evidence set $E = e_1, ..., e_q$ for each abstract a.

Label Prediction

For the label prediction, the inputs are claim c and evidence set $E = e_1, ..., e_q$, the aiming output is the claim label y.

The claim c and evidence e_i form the sentence pair representation H^i with BERT model.

$$H^i = BERT([CLS], c, [SEP], e_i, [SEP]) \tag{2.9}$$

The sentence pair representation H^i contains tokens from both the claim and evidence e_i. The probability of claim label y is calculated.

$$p(y|c, E) = KGAT(H^1, ..., H^q) \tag{2.10}$$

The claim label y with the largest probability is the obtained output.

Continuous In-Domain Training

The continuous in-domain training method that has been used by the SciKGAT model including rationale prediction based training and mask language model based training. For the low-resource fact checking mission like COVID-FACT, as the development of medical method and knowledge, the corpus will also expand. Then the pre-trained language models will expire. The continuous in-domain training method gives us a solution for solving this kind of problem.

The first method is rationale prediction based training. The BERT model is trained on SCIFACT. The cross-entropy loss between y_r and ${y_r}^*$ is minimized, where ${y_r}^*$ is the golden rationale prediction label of claim c and evidence e.

$$L_r(c, e) = CrossEntropy(p(y_r|c, e), {y_r}^*) \tag{2.11}$$

Therefore, we obtained a supervised in-domain language model BERT-RP for the abstract retrieval and rationale selection step. The state-of-art model KGAT is trained and implemented for the fact verification step.

The second method is mask language model based training. Some tokens are replaced with [MASK]. The model aims to generate appropriate tokens to fill them in. By doing so, the model can better understand the semantics of new corpus, so that the new terminologies can be learnt.

2.4.3 Existing Results Illustration

We consider the existing Sentence-Level correctly labeled scores and Abstract-Level correctly rationalized scores on VERSCI and SciKGAT. The scoring methods have been discussed in Section 2.3.3. We can see that on the Sentence-Level, SciKGAT exceeds the performance of VERSCI. The Precision on dev set is 74.36 and on test set is 61.15. The Recall on dev set is 39.62 and on test set is 42.97. And the overall F_1 score on dev set is 51.69 and on test set is 50.48. For the Abstract-Level evaluation, the SciKGAT also exceeds the VERSCI except for the Recall on dev set. The higher Precision on dev set is 84.26 and on test set is 76.09. The higher Recall on dev set is 46.41 and on test set is 47.30. And the F_1 score is 57.41 on dev set and 58.33 on test set. We can conclude that the SciKGAT model improved the performance with around 10% on the F_1 score on both the Sentence-Level and the Abstract-Level with KGAT and continuous training. And there is still a large space for the improvement of the system performance.

Standard	Sentence Level			Abstract Level		
Dev Set	Precision	Recall	F_1	Precision	Recall	F_1
VERSCI(RoBERTa-large)	46.51	38.25	41.98	53.30	**46.41**	49.62
SciKGAT(Full)	**74.36**	**39.62**	**51.69**	**84.26**	43.54	**57.41**
Test Set	Precision	Recall	F_1	Precision	Recall	F_1
VERSCI(RoBERTa-large)	38.6	40.5	39.5	46.6	46.4	46.5
SciKGAT(Full)	**61.15**	**42.97**	**50.48**	**76.09**	**47.30**	**58.33**

Table 2.10: Existing Scores for SCIFACT Task

2.5 Comparison between FEVER and SCIFACT Task

The SCIFACT dataset and FEVER dataset share similarities and hold differences. We will illustrate this in detail, and make conclusion on how to use these specialities for the model and training method selection regarding SCIFACT task.

2.5.1 Similarities

Firstly, the pipelines are similar. The document retrieval from FEVER is similar to abstract retrieval from SCIFACT. Second step in FEVER, sentence-level evidence selection, show common ground with rationale selection in SCIFACT. And claim verification in FEVER is as same as label prediction in SCIFACT.

Secondly, the label classification is similar: both include SUPPORT, REFUTE and NOT ENOUGH INFO.

Thirdly, for both tasks the maximum number of predicted evidence sentences are limited. For FEVER task, Alammar, 2018a limit the maximum number of rationale sentences prediction to five. For SCIFACT task, the number is limited to three.

2.5.2 Differences

We compare the two datasets to mark down differences from the perspectives of dataset size, dataset type, evaluation method, models implemented, training method, existing result and application field.

Dataset Size

The large-scale FEVER dataset contains 165,447 claims for the FEVER task while the small-scale SCIFACT dataset contains 1409 claims for the SCIFACT task.

Dataset Type

The source of FEVER dataset are Wikipedia articles which is common knowledge, while the source of SCIFACT dataset are scientific articles published on authoritative scientific journals. Thus comparing between the two, the claim verification for SCIFACT dataset requires more modeling capabilities including: science background, directionality, numerical reasoning, cause and effect and coreference[7].

The claims in SCIFACT dataset are natural while the claims in FEVER dataset are synthetic. Since the claims in SCIFACT are extracted from citation sentences which take place naturally in scientific articles (Wadden et al., 2020) while in FEVER dataset the claims are created by annotators from Wikipedia articles.

Evaluation Method

For the FEVER task, the evaluation is conducted for each sub-task and full pipeline. For the sub-tasks it includes fully supported and oracle accuracy for document retrieval, precision,recall and F1 for sentence-level evidence selection and label accuracy and FEVER score for the claim verification. For the full pipeline the evaluation is based on the FEVER score.

For the SCIFACT task, the full pipeline evaluation is conducted on the abstract-level and sentence-level. For abstract level, it includes the precision, recall and micro-F1 score for label only and label+rationale. For sentence level, it includes the precision, recall and micro-F1 score for selection only and selection+label. On abstract level, the evaluation is the "Open" FEVER style.

Models Implemented

There are certainly a lot more models being carried out on the FEVER dataset than on the SCIFACT dataset. Existing models on the SCIFACT dataset including TF-IDF, BERT and KGAT are first implemented on the FEVER task and later applied to the SCIFACT task.

Training Method

For the FEVER task we train the models in order to minimize the pointwise or pairwise loss with the supervision of FEVER dataset. Meanwhile on SCIFACT task, we train the models and minimize cross-entropy loss. The supervision dataset is the SCIFACT dataset or the concatenation of the SCIFACT and FEVER dataset for a larger training range. For SCIFACT task, the continuous in-domain training is implemented to adapt open domain fact extraction and verification on other medical corpus like COVID-19 Open Research Dataset Challenge[8].

Existing Result

Application Field

The FEVER task provides the testbed for applications like fake news detection and medical claim verification, which require the depth of language understanding. And with the basis provided by FEVER task, the SCIFACT task focuses on applications like scientific fact-checking and verifying real world claims related to medical crisis such as COVID-19. These applications require the depth of scientific language understanding.

[7] The specific explanations for these terms are illustrated in A.1.

[8] https://www.kaggle.com/allen-institute-for-ai/CORD-19-research-challenge

2.5.3 Conclusion

We choose to study on SCIFACT dataset because there are blank spaces on research related fact verification. It contains more complicated sentence structure and requires more experts for dataset creation. Due to the similarities between SCIFACT dataset and FEVER dataset, models implemented on FEVER dataset with good result can be implemented on SCIFACT dataset when meeting the particularities of SCIFACT task.

Chapter 3

Models

In this chapter, we first discuss about the preliminary models that we need for the Fact Verification task. The preliminary models include Transformer and BERT. The most important part of Transformer is its attention mechanism. BERT inherited the encoder part of Transformer and added the bidirectional training method. Then we introduce the state-of-art models from VERSCI (Wadden et al., 2020). We first bring in the classic TF-IDF method and retrieve the top relevant abstracts. Next we use the BERT-style models SCIBERT and RoBERTa-large to retrieve evidence sentences from the golden abstracts. In the end, we introduce the Transformer-XH model based on the Transformer structure and apply it on the label prediction step for SCIFACT task.

3.1 Preliminary Model: Transformer

Transformer is developed by Vaswani et al., 2017 and it is broadly implemented on many Nature Language Processing tasks. The model takes sequence of words as input and gives the output regarding the specific task. We perceive in Figure 3.1 that the Transformer consist of Encoders and Decoders and the concatenation between them. Regarding the work of Vaswani et al., 2017, there are six encoders piled up together and also there are six decoders as symmetric.

The encoders share the same structure but their parameters are different. Each encoder contains two parts: Self-attention layer and Feed Forward layer. The input sentence first go through the self-attention layer, which supports each word to pay attention to words from rest of the sentence. The output of the self-attention layer is then transferred to the feed-forward neural network. The feed-forward neural network stays same for each word.

The decoder includes two self-attention layers and one feed-forward layer. The extra self-attention layer is functioned to make the decoder acknowledge the related sectors of the input sentence. But in the Fact Verification task, the encoder part is what we take and use, so we don't go into details about the decoder part. The details about encoder part is illustrated in Figure 3.2.

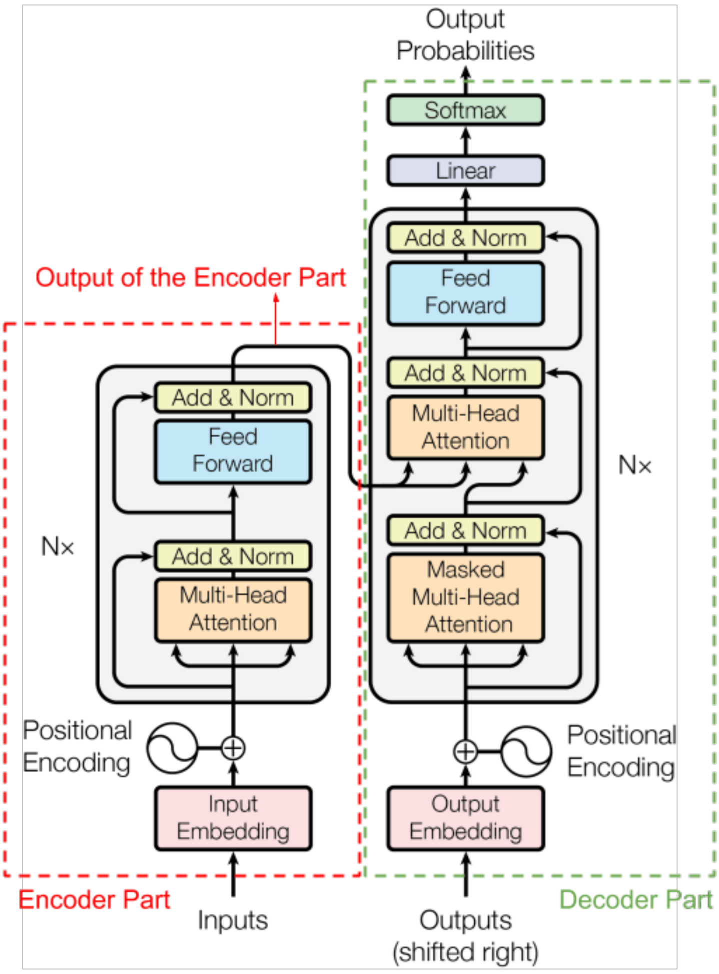


Figure 3.1: Transformer Model (Vaswani et al., 2017). For fact verification task, we focus on the encoder part.

3.1.1 Word Embedding

Just like most of the NLP models, we first transform every input words to word vectors through word embedding. We fix every word into a 512-dimension vector. Word embedding only takes place at lowest part of the encoders. All the encoders take one vector list with all vectors at size 512. At the foot encoder, the input is word vector. And at the other encoders, the input is the output from the previous encoder, which is also a vector list. We can adjust the size of the vector list when setting up the hyper parameters. It is usually the longest input sentence in our training set.

After performing word embedding, every word goes through the self-attention layer alongside the feed-forward layer, which is the core feature of the Transformer model. At the input sequence part, every word has its own path to flow into the encoder. At the self-attention layer part, there

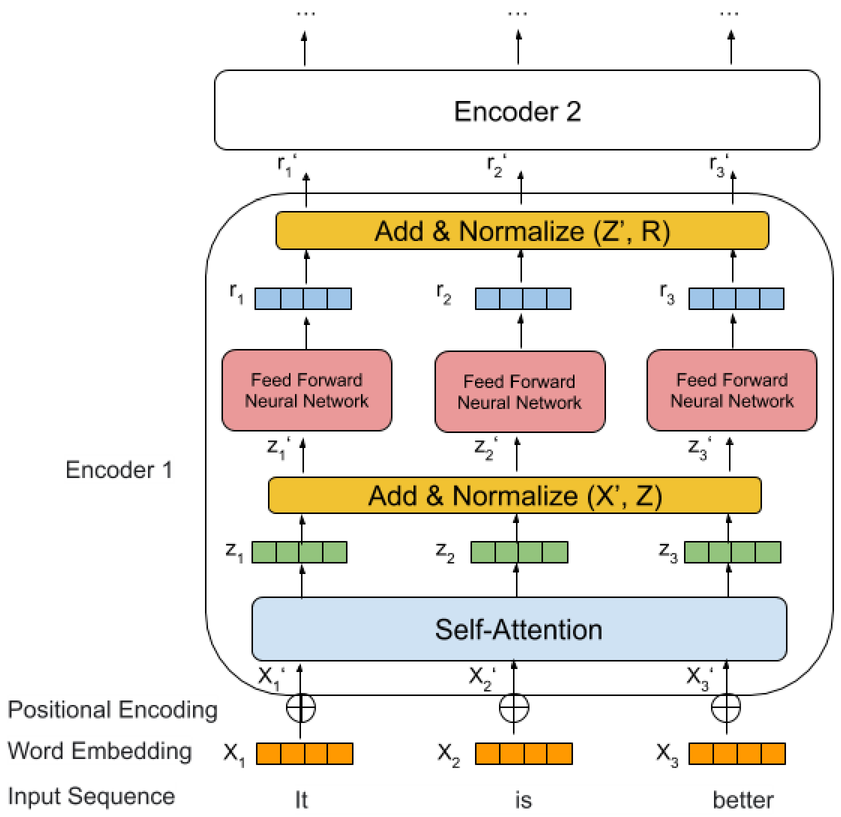


Figure 3.2: The Detailed Encoding Procedure with Tensors.

are dependencies on these paths. While on the feed-forward layer there is no dependency for the paths. Therefore at the feed-forward layer we are able to execute various paths in parallel. We show the procedure in Figure 3.2

3.1.2 Attention

As we illustrated before, an encoder takes a vector list as input, and the vectors are forward to the self-attention layer. Then the output from the self-attention layer is forwarded to the feed-forward neural network. The output of the precious encoder is forwarded to the next encoder. When the model is processing a word from the input, the self-attention mechanism find attention on every word of the input sequence to help the model to better encode the word. And the attention mechanism put the understanding of all the related words into the word that we are proceeding.

Self-Attention

We initially produce three additional vectors from the input sequence as depicted in Figure 3.3. That is to say, we generate a Query, a Key and a Value for each word. These three vectors are generated through the multiplication of word embedding and weight matrices. We can observe that the new vectors are smaller on the dimension perspective comparing to the input and output vectors. The new vectors are 64-dimensions and the input and output vectors are 512-dimensions. But the dimension is just a choice based on the architecture.

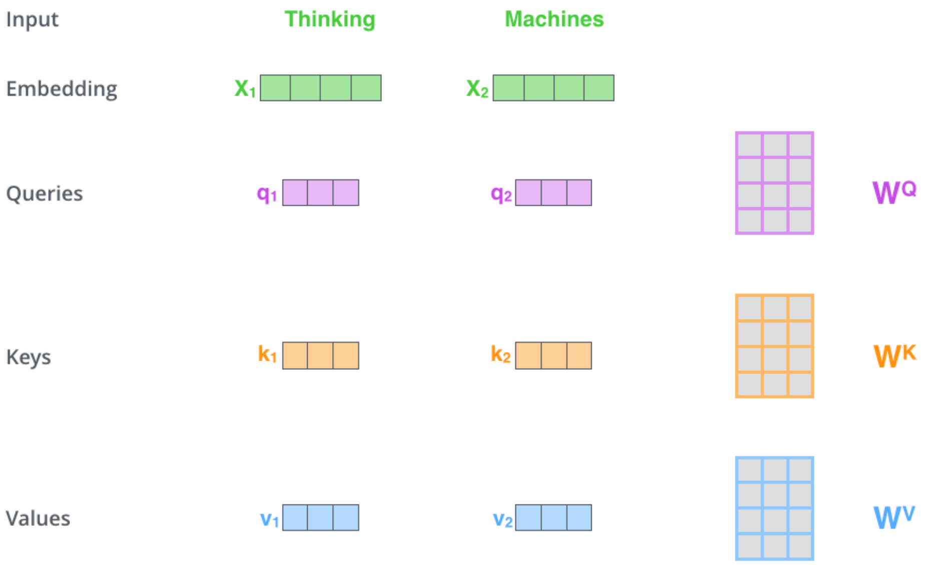


Figure 3.3: Transformer Self-Attention Vectors (Alammar, 2018b).

We then compute the score. For instance, if we would like to compute the self-attention score for the i-th word in the sentence. We need to have the scoring of every words on the i-th word. These scores determine how we attach importance to other parts of the sentence when encoding the i-th word. We compute the scores through dot product of the key vector regarding every word and the query vector regarding the i-th word.

The third step is to make the gradient more stable through dividing the score with the square root of dimension as $\sqrt{64} = 8$.[1] Then we implement the softmax function to normalize the score of every word, and the summation of score for every words is 1. The normalized score reflects the contribution of each word to the i-th word. The i-th apparently will have the highest score.

In the end, we multiply the value vector with the normalized score to obtain the weighted value vector. The goal here is to increase the attention on semantically related words and reduce the attention on non-related words. Then we vector sum the adjusted value vector and obtain self-attention layer output.

[1] Other values can also be used, 8 is just a default.

We calculate the attention function on a list of query vectors at the same time. The query vectors are packed together into a matrix Q. The key vectors and value vectors are also packed together into matrices K and V. We calculate the output Z in Equation 3.1 (Vaswani et al., 2017).

$$Attention(Q, K, V) = softmax(\frac{Q \cdot K^T}{\sqrt{d_k}}) \cdot V \tag{3.1}$$

The attention is calculated through dot-product (matrix multiplication) as illustrated in Figure 3.4, and we also call it Scaled Dot-Product Attention.

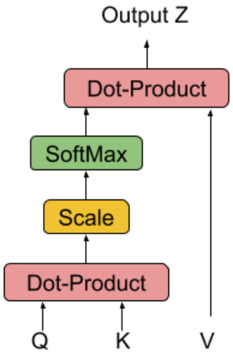


Figure 3.4: Scaled Dot-Product Attention (Vaswani et al., 2017).

Multi-Head Attention

We completed the self-attention layer with ”multi-head” attention mechanism. The feature of attention part is elevated from two aspects. **First**, it expanded the capability of the architecture to focus on divergent locations. At the example above, although every word is illustrated in vector z_i, but it will be dominated by the word itself. **Second**, it transmits multiple subspace representations to the self-attention layer. Every head is with its matrix sets and is initialized randomly. After the training, every head is used for projecting the input word embedding (or vector from lower encoder/decoder) into different subspace representations.

For ”multi-head” attention method, we project query vector, key vector and value Vector linearly to every head and keep the matrices at each head independent. As a result, we get multiple Z matrices. Since we only need one Z matrix for the feed forward layer, we then compress these matrices into one matrix through concatenation. And later multiply them with an additional weight matrix W^O. Weight matrix W^O is trained jointly during training process. The visualisation for the multi-head attention mechanism is depicted in Figure 3.5.

We calculate the Multi-Head Attention in Equation 3.2 and 3.3 (Vaswani et al., 2017).

$$MultiHead(Q, K, V) = Concat(head1, ..., head_h)W^O \tag{3.2}$$

$$head_i = Attention(QW_i^Q, KW_i^K, VW_i^V) \tag{3.3}$$

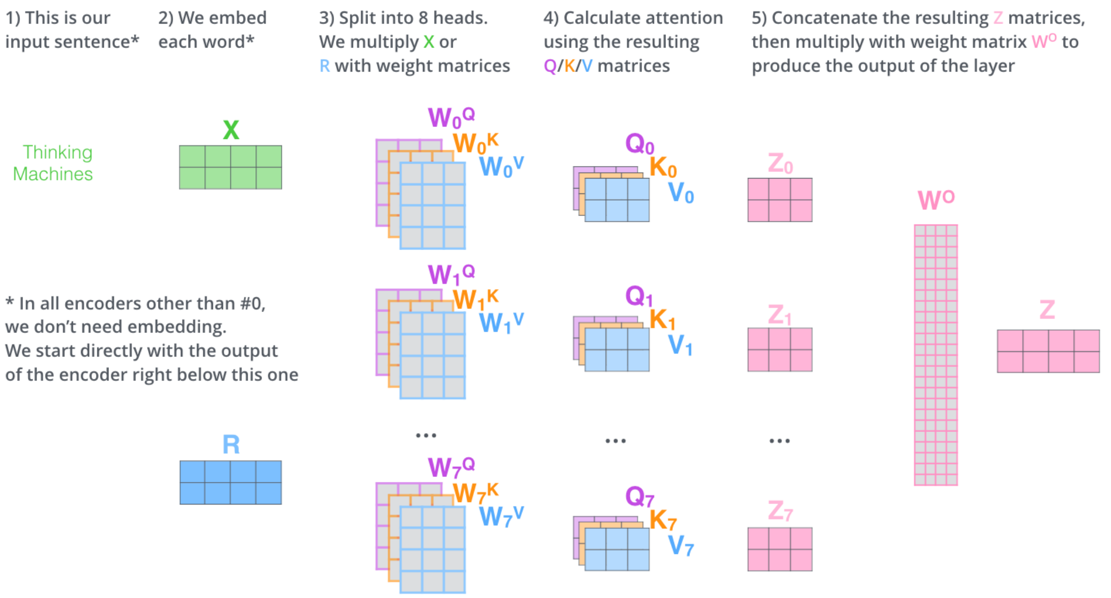

1) This is our input sentence*
2) We embed each word*
3) Split into 8 heads. We multiply X or R with weight matrices
4) Calculate attention using the resulting Q/K/V matrices
5) Concatenate the resulting Z matrices, then multiply with weight matrix W^O to produce the output of the layer
Thinking Machines
X
* In all encoders other than #0, we don't need embedding. We start directly with the output of the encoder right below this one
R
W0Q W0K W0V
W1Q W1K W1V
...
W7Q W7K W7V
Q0 K0 V0
Q1 K1 V1
...
Q7 K7 V7
Z0
Z1
...
Z7
WO
Z

Figure 3.5: Visualization for the Multi-Head Attention (Alammar, 2018b).

3.1.3 Other Parts of the Encoder

Feed-Forward Network

The position-wise feed-forward neural network includes two linear transformations. The ReLU activation is used between them (Vaswani et al., 2017).

$$FFN(x) = max(0, xW_1 + b_1)W_2 + b_2 \tag{3.4}$$

Positional Encoding

Transformer model embedded a position vector for each input word to illustrate understanding on the word order or the distance between different words.

The Residual Module

Each sub layer in the encoder is connected with a residual module following with a normalization.

3.1.4 Advantages

After the illustration of the model structure. We also introduce the motivation for using self-attention.

- Transformer reduced the total complexity of computation per layer.
- Transformer can paralyse an amount of computation, thus minimize number of sequential operations.

- Transformer extended the path length between long-range dependencies in the network. Learning long-range dependencies is a key challenge in many sequence transduction tasks (Vaswani et al., 2017).

3.1.5 Conclusion

Transformer is a sequence transduction model based entirely on attention. The recurrent layers most commonly used in encoder-decoder architectures are replaced with multi-headed self-attention (Vaswani et al., 2017). The model is the basis of BERT, which is a step forward model for the Fact Verification task. We carried out the BERT style model on the sentence retrieval step. Transformer is also the basis of Transformer-XH model implemented on the label prediction, which added extra hop onto it for multi-evidence reasoning.

3.2 Preliminary Model: BERT

The Bidirectional Encoder Representations from Transformers (BERT) model relies on WordPeice embedding (Wu et al., 2016) and Transformer networks (Vaswani et al., 2017). The encoder part of Transformer contributes a significant part to BERT. And BERT is also the basis model for Roberta-large (Y. Liu et al., 2019) and SciBERT (Beltagy et al., 2019). And these two models take the BERT and train it with different corpus to make the best use of it.

3.2.1 Model Architecture

The BERT model includes BERT BASE version and BERT LARGE version. The BERT BASE version has the equivalent size as OpenAI Transformer for comparing their performance. The BERT LARGE version is a very big model and has been given many best results on different tasks. The basic component of BERT is the encoder of Transformer, which is also called the Transformer Block. For BERT BASE, there are 12 Encoders piled up, and for BERT LARGE, the number is 24. The Feed-Forward Neural Network in the encoder is with 768 hidden layer neurons for the BASE version and 1024 for the LARGE version. The number of attention heads is from 12 to 16. The amount of parameters implemented in BERT exceeds the reference configuration parameters of Transformer. The model architecture of BERT is shown in Figure 3.6.

The input is one sentence or sentences encoded in a single sequence. The first token of the sequence is always the special token [CLS], the abbreviation [CLS] shows it's purpose for classification. The sentences are separated by the special [SEP] symbol, the abbreviation [SEP] shows it's purpose for separate and concatenate sentences to perform task like whether sentence A is similar to sentence B. The encoding process of BERT is same as Transformer and the structure of the BERT is inherited from the Transformer. We input a sequence and calculate the output layer by layer. For every layer the Self-Attention and the Feed Forward Neural Network are implemented. But at the output of the BERT model, the structure is different. The output vector at every position is with the size of the hidden layer. For sentence retrieval and label classification, when a BERT-style model is implemented, the focus is on the first position output [CLS].

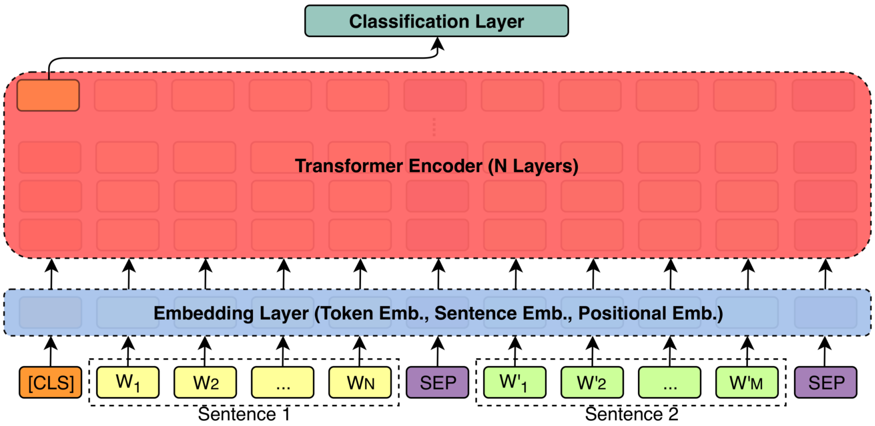

Classification Layer
Transformer Encoder (N Layers)
Embedding Layer (Token Emb., Sentence Emb., Positional Emb.)
[CLS]
W1
W2
...
WN
SEP
W'1
W'2
...
W'M
SEP
Sentence 1
Sentence 2

Figure 3.6: Model Architecture of BERT (Soleimani et al., 2019).

3.2.2 Pre-Training BERT

The word embedding has become a very important part of the NLP model. Word2Vec algorithm uses a group of fixed dimension vectors to represent the words. The calculation method can capture the semantics of words and the relationship between words. But the problem is that for the pretrained word embedding model, each word has only one unique and fixed vector form, which cannot be changed with contextual relations. ELMo (Peters et al., 2018) solves the contextual problem with Bi-LSTM, which can interpret not only the next word but also the previous word.

Although the good performance of Transformer makes some researchers believe that it can be the substitution of LSTM structure. And the advantage is that the Transformer is better at dealing with long-term dependencies compared with LSTM. But Transformer is a forward-trained language model and lacks of a bidirectional structure. Therefore the BERT model implemented Masked Language Model together with the encoders from Transformer and made the training bidirectional. We mask the word that need to be predicted because the mechanism of Self-Attention will certainly predict the word without masking. The Next Sentence Prediction task is also used to pre-train the model as we need to concatenate claim with sentence in the Natural Language Interface task as depicted in Figure 3.7.

3.2.3 Conclusion

On the basis of deep unidirectional architectures of Transformer, the BERT model further extended it into bidirectional architecture. The pre-trained model can accomplish good results in many NLP tasks including Fact Verfication.

3.3 Models on Abstract Retrieval and Rationale Selection

The source code link for SciKGAT is the KGAT model not including SCIFACT dataset and the sentence retrival results is not released yet, so we instead reproduce the BERT-based retrieval results from baseline model.

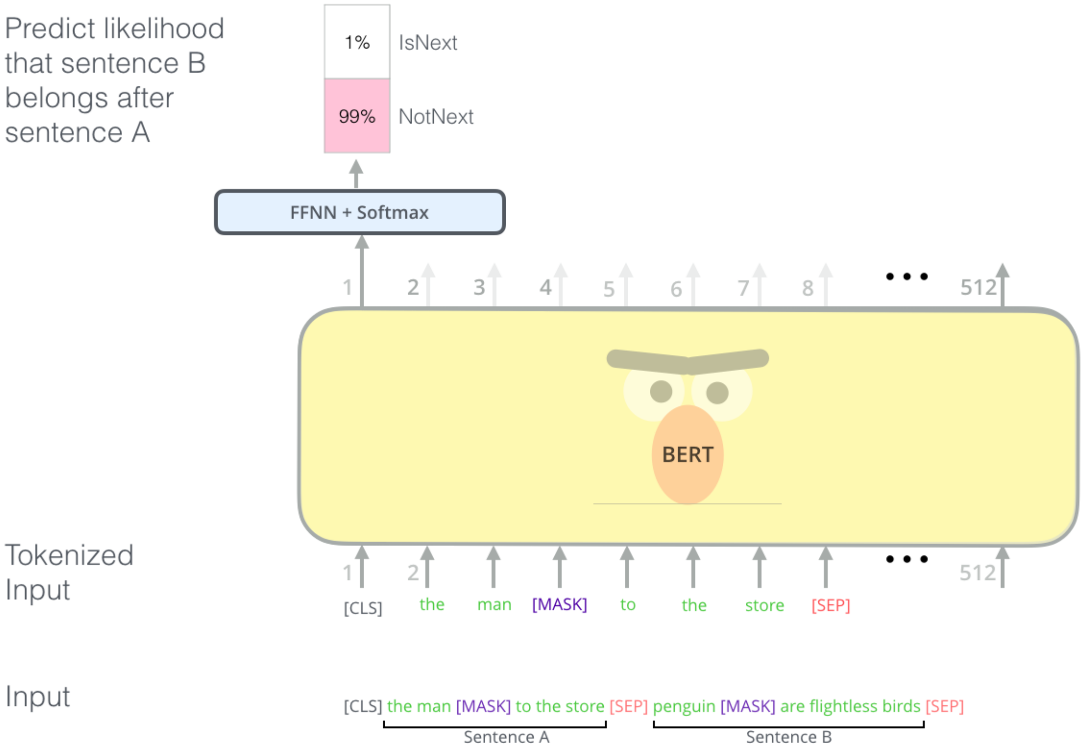

Predict likelihood that sentence B belongs after sentence A
1%
IsNext
99%
NotNext
FFNN + Softmax
1
2
3
4
5
6
7
8
...
512
BERT
Tokenized Input
[CLS]
the
man
[MASK]
to
the
store
[SEP]
Input
[CLS] the man [MASK] to the store [SEP] penguin [MASK] are flightless birds [SEP]
Sentence A
Sentence B

Figure 3.7: Pre-Training BERT with Masked Language Model and Next Sentence Prediction (Alammar, 2018a)

.

We first use TF-IDF to locate the most relevant abstracts and then use Roberta-large model to find the evidence rationales.

3.3.1 TF-IDF for Abstract Retrieval

TF-IDF is the combination of Term Frequency (TF) and Inverse Document Frequency (IDF).

The Term Frequency represents the frequency of phrases, which means the importance of a word inside a document is related to how often it appears. When a word or a phrase appears more often, then it is more important. TF is used in the early age of search engine development, and some people insert a lot irrelevant top words of searching list in order to make their website rank higher. So IDF is introduced to tackle this problem.

The Inverse Document Frequency means that when the more broadly exist terms are less important. This also means that the importance of a term is inversely proportional to its generality. IDF makes increased the precision of the system as the words that appear fewer have greater reference meaning.

$$idf(t, D) = log \frac{N}{1 + |\{d \in D : t \in d\}|} \tag{3.5}$$

where

- N is the total number of documents from corpus D,
- $|\{d \in D : t \in d\}|$ is the total number of documents where the term t exists in,
- and the 1 in the denominator is for avoiding division-by-zero.

The Term Frequency measures the relevance between the phrase and the document. And the Inverse Document Frequency measures the relevance between the phrase and all the documents. In other words, the two are a bit like the relationship between the part and the whole. We multiply the two to get the final importance expression in Equation 3.6, which means that TF-IDF is an algorithm used to calculate the importance of a phrase in a document.

$$tfidf(t, d, D) = tf(t, d) \cdot idf(t, D) \tag{3.6}$$

And with TF-IDF, we get k abstracts with highest similarity to the claim.

3.3.2 BERT-Style Models for Rationale Selection

After retrieving the relevant abstracts, we train BERT-style Model and use it for rationale selection. In the VERSCI model, the RoBERTa-large performs well on both the rationale selection and label prediction task. And the SCIBERT is a bit better on Rationale Selection than RoBERTa-large. So we reproduced the training process for both of the models on SCIFACT dataset and implemented them to retrieve the evidence sentences.

SCIBERT

The SCIBERT model (Beltagy et al., 2019) take the pretrained BERT-BASE model with the same parameter settings and then perform training on different NLP tasks including: Named Entity Recognition (NER), PICO Extraction (PICO), Text Classification (CLS), Relation Classification (REL) and Dependency Parsing (DEP). In the end we have the pretrained SCIBERT model which outperforms the BERT-BASE model on scientific tasks.

RoBERTa-large

A replication study on BERT pretraining with careful evaluation of the effects of hyperparmeter tuning and training set size is performed by Y. Liu et al., 2019. And they discovered that BERT was significantly undertrained, so the improved version called RoBERTa is introduced. The performance of RoBERTa model is equivalent to or better than the original BERT model.

The modifications of RoBERTa on BERT include:

- training with longer time, bigger batches, longer sequences and more data;
- erasing the next sentence prediction purpose;

- adjusting the masking pattern dynamically.

The pretrained RoBERTa-large model is modified on the pretrained BERT LARGE model and improved the performance.

Conclusion

The training method for rationale selection are illustrated in Section 2.4.1. In the end we retreived the evidence sentences from the corpus. Then we can construct the evidence graph which later can be used on Transformer-XH model for label prediction.

3.4 Model on Label Prediction

3.4.1 Sequential Transformers

Transformer

Suppose we have a sequence of text tokens $X = \{x_1, .., x_i, ..., x_n\}$ being represented by contextualized distributed representations $H = \{h_1, ..., h_i, ..., h_n\}$ with transformers (Vaswani et al., 2017). Multiple stacked self attention layers are implemented to convert the sequence X into $\{H^0, H^1, ...H^l, ...H^L\}$. The attention mechanism calculates the $l - th$ layer output H^l with the output from the $(l-1)th$ layer H^{l-1} as input.

$$H^l = softmax(\frac{Q \cdot K^T}{\sqrt{d_k}}) \cdot V \tag{3.7}$$

, where $Q^T; K^T; V^T = W^q \cdot H^{l-1}; W^k \cdot H^{l-1}; W^v \cdot H^{l-1}$ represent three projections Query (Q), Key (K), and Value (V) on the input H^{l-1}.

The token ${h_i}^l$ in output H^l is shown in Equation 3.8.

$${h_i}^l = \sum_j softmax_j(\frac{{q_i}^T \cdot k_j}{\sqrt{d_k}}) \cdot v_j \tag{3.8}$$

We first calculate the i-th token's attention to all tokens (represented as j-th token) in the sequence X. We then implement the normalized attention weights and combine the token value v_j into ${h_i}^l$. Multiple attentions can be used in one Transformer layer and concatenated as multi-head attention (Vaswani et al., 2017). Deep networks are formed with this architecture and lead to large pre-trained Transformer models (Devlin et al., 2019 ;Y. Liu et al., 2019).

Transformer-XL

However, the attention is calculated over all token pairs in sequence X for the Transformer. Thus in the case of long text sequences, the attention is hard to scale. The variation of Transformer namely Transformer-XL (eXtra Long) (Dai et al., 2019) separate longer texts to tackle this problem. For instance, the long sequence X is divided into a sequence of text segments

$\{X_1, ..., X_\tau, ..., X_\zeta\}$. The attention in Equation 3.9 propagates the information between neighbouring text segments.

$$\tilde{H}_\tau^{l-1} = [Freeze(H_{\tau-1}^{l-1}), H_\tau^{l-1}] \tag{3.9}$$

The representation of the previous segment $H_{\tau-1}^{l-1}$ is concatenated to the current segment H_τ^{l-1}, thus the new current segment representation $\tilde{H}_\tau^{l-1}$ contains information from previous segment. And this is referred to as segment level recurrences.

$$\tilde{H}_\tau^{l-1} = [Freeze(H_{\tau-1}^{l-1}), H_\tau^{l-1}] \tag{3.10}$$

The new current segment representation is updated in the attention mechanism for Transformer-XL (eXtra Long).

$$\tilde{Q}^T; \tilde{K}^T; \tilde{V}^T = W^q \cdot H_\tau^{l-1}; W^k \cdot H_\tau^{l-1}; W^v \cdot \tilde{H}_\tau^{l-1} \tag{3.11}$$

Due to the attention mechanism which includes previous segment, the Transformer-XL can model long text data recurrently as a sequence of text segments.

In some cases, the text segments are not structured in a linear sequence, so the performance of Transformer-XL is restricted. Thus we need the Transformer-XH (eXtra Hop) (Zhao et al., 2020) attention to further address this challenge.

3.4.2 Transformer-XH

Model Structure

We introduce the structured text data for Transformer-XH as a non-linear set of linear sequences. It contains a group of nodes $\mathcal{X} = \{X_1, ..., X_\tau, ..., X_\zeta\}$. Each node represents a text sequence and corresponds to an edge matrix E. The edge matrix E represents the links between the nodes.

We intend to acquire the contextualized distributed representations $\mathcal{H} = \{H_1, ..., H_\tau, ..., H_\zeta\}$ to combine the local information and the global information. Therefore, the Transformer-XH includes two attention mechanisms: in-sequence attention for the local information in each sequence X and eXtra Hop attention for the global information on all structured text $\{\mathcal{X}, E\}$.

The in-sequence attention helps token i at layer l to collect information from other tokens inside the same text segment τ, which is the same method as in sequential Transformers (Section 3.4.1).

$$h_{\tau,i}^l = \sum_j softmax_j(\frac{q_{\tau,i}^T \cdot k_{\tau,j}}{\sqrt{d_k}}) \cdot v_{\tau,j} \tag{3.12}$$

The eXtra Hop attention contains an "attention hub" which is the first token [CLS] in each sequence. The attention hub is joint with other hub tokens if their nodes are connected. Let's assume that there is an edge ($e_{\tau\eta} = 1$) between the τ-th text sequence and the η-th text sequence in layer l, meanwhile the τ-th text sequence joins in the η-th text sequence. The global representation $\hat{h}^l_{\tau,0}$ is calculated in Equation 3.13.

$$\hat{h}^l_{\tau,0} = \sum_{\eta;e_{\tau\eta}=1} softmax_\eta(\frac{\hat{q}^T_{\tau,0} \cdot \hat{k}_{\tau,0}}{\sqrt{d_k}}) \cdot \hat{v}_{\tau,0} \tag{3.13}$$

Node τ calculates the attention weight on the neighbouring node η through hop query $\hat{q}_{\tau,0}$ and key $\hat{k}_{\tau,0}$ (Zhao et al., 2020). Furthermore, the attention weights are used to combine the value $\hat{v}_{\tau,0}$ of neighbouring nodes and form the global representation $\hat{h}^l_{\tau,0}$.

Later, we combine the two attention mechanisms and constitute the new representation of layer l.

$$\tilde{h}^l_{\tau,i} = \begin{cases} Linear([h^l_{\tau,0}], \hat{h}^l_{\tau,0}]), & \text{if i = 0} \\ h^l_{\tau,i}, & \text{otherwise} \end{cases} \tag{3.14}$$

We can conclude from Equation 3.12 and 3.14 that the non-hub tokens also get in touch with the hop attention from the previous layer.

Model Explanation

In Transformer-XH, each layer is a single-step information propagation with edges E. As a result, a Transformer-XH model with L layers can have access to information within L hops further. This can be illustrated from the following example in Figure 3.8. Node d_1, d_2 and d_3 are one group of nodes. d_3 obtain information from d_1 with the hop attention from d_1 to d_3, which is happening on the 2-nd layer. And d_1 obtain information from d_2 with the hop attention from d_2 to d_1, which is happening on the 1-st layer. Thus the information from up to 2 hops away is obtained.

Model Properties

Thanks to the structure of the model, Transformer-XH can validly model original structured text data from the following aspects. **First**, the information can be propagated through edges. **Second**, the importance of the information propagated through edges are calculated with hop attention weights. **Third**, the in-sequence attention and eXtra Hop attention are combined in the attention mechanism. As a result, the $\mathcal{H}$ representations can stand for the fine differences of structured text, which is very useful in the complicated reasoning task of natural language inference like FEVER and SCIFACT.

3.5 Application to SCIFACT

3.5.1 Abstracts Retrieval with TF-IDF

We take in the claim and corpus of the SCIFACT dataset and use TF-IDF to select the 3 most relevant abstracts. The unigrams and bigrams are extracted. In this step we extract the abstracts for the training set because we want to feed the label prediction model with the evidence sentences retrieved for the claims labeled as NOT ENOUGH INFO. We also performed the accuracy test

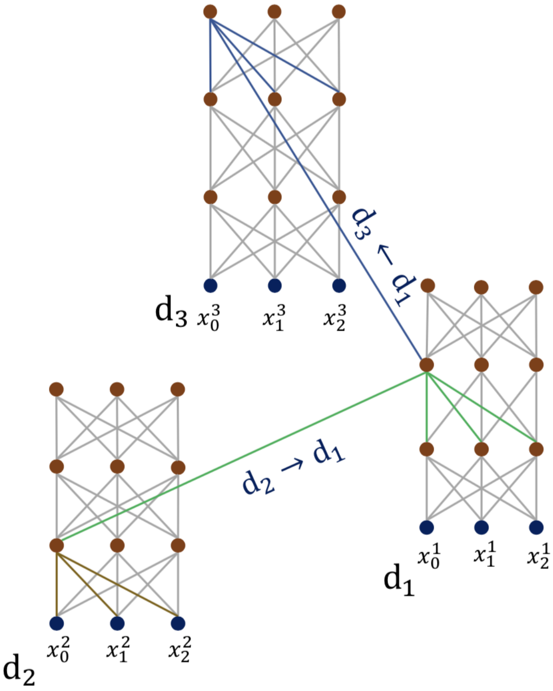


Figure 3.8: Hop Attentions on the Path from d_2 to d_1 to d_3 (Zhao et al., 2020).

on the training set and the accuracy for at least one abstract is retrieved correctly is 0.8653. The accuracy of retrieving all the abstracts correctly is 0.8356. We also performed abstract retrieval with TF-IDF on the dev set and the result for that is shown in Section 4.3.1.

3.5.2 Rationale Selection with BERT

The next step is to extract evidence sentences from the selected abstracts. In the paper of Wadden et al., 2020 they trained SCIBERT, BioMedRoBERTa, RoBERTa-base and RoBERTa-large on the FEVER, UKP Snopes, SCIFACT and FEVER+SCIFACT dataset. The ones give the better results on rationale selection are SCIBERT or RoBERTa-large trained on the SCIFACT. The reasoning behind the selection of dataset is that SCIFACT dataset contains scientific vocabularies which are not in other corpus like FEVER, so the additional training on other datasets doesn't help. Therefore we reproduced the process of SCIBERT and RoBERTa-large and obtained the evidence sentences.

We take retrieved evidence abstracts for the training set and use RoBERTa-large model to perform rationale selection in order to obtain the sentences for evidence graph construction for the training at the label prediction step. We evaluated the result accuracy of the concatenation of step one and two. And the results we have on the training dataset are 0.1254 for Precision, 0.5941 for Recall and 0.2071 for F1. As our goal is to retrieve training sentences for the third step, so the results are acceptable. The evaluation of Rationale Selection on dev set is illustrated in Section 4.3.2.

3.5.3 Label Prediction with Transformer-XH

In this section we describe construction of the evidence graph with evidence pieces including evidence sentences and abstract titles from the reliable corpus. Later we can apply Transformer-XH to perform reasoning on the evidences.

Evidence Graph Construction

We make the evidence rationale sentence retrieval step consistent with the previous methods. For SCIFACT dataset, the golden label SUPPORT or CONTRADICT is written for each evidence sentence and for each claim the golden evidence sentences are all SUPPORT or all CONTRADICT . So we extracted the evidence sentence labels to one claim label stated for the claim as SUPPORTS or REFUTES. We give the claim label as NOT ENOUGH INFO to the claims with no golden rationales.

We construct the evidence graph. For claims with golden rationales (SUPPORTS and REFUTES) we directly put the golden evidence sentences in the training and dev set and set the label for the node as 1, which represents golden evidences. And for claims with no golden rationales (NOT ENOUGH INFO), we take the rationales from the rationale selection with RoBERTa-large as sentence encoder and set the parameter to K2, which means 2 sentences to select for each abstract. In the abstract retrieval step we choose the 3 abstracts with highest TF-IDF, so we have $2x3 = 6$ sentences for each NOT ENOUGH INFO claim, the node labels are set to 0, which represents predicted evidences.

The evidence graph for one example SCIFACT claim is depicted in Figure 3.9. We can observe that the JSON structure contains five elements: $qid, question, label, node$ and $edge$. First, we fill in the "qid" with the claim id in the SCIFACT dataset. Second, we constructed "question" with the content of the claim. Third, we formed the "label" with the golden claim label. And then we acquire the group of nodes with 6 nodes contained specifically in this case. The nodes are illustrated with the "node_id", "name", "content", "sent_num" and "label". Last, we created the empty edge vector for the future upgrade while training.

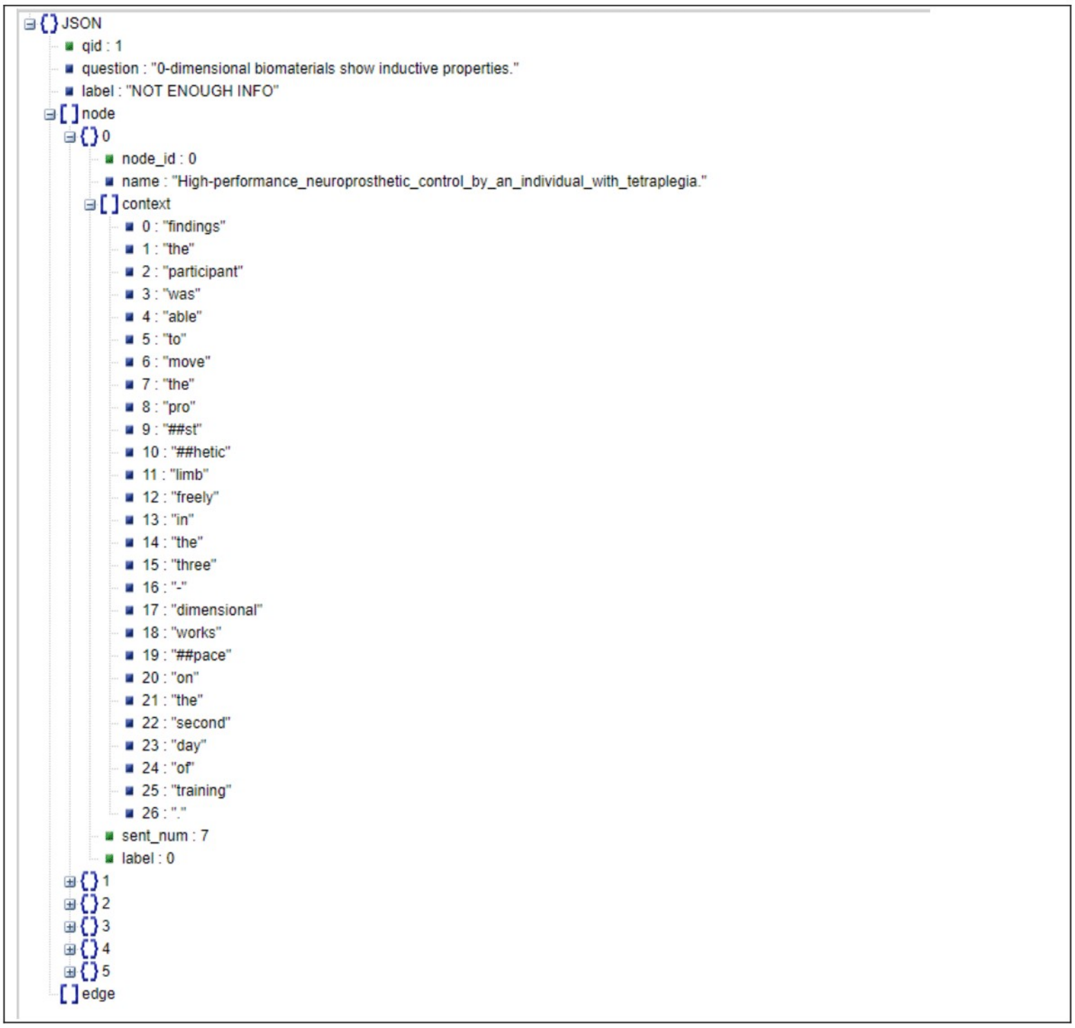
Concretely for the first node in our example, we have the "node_id" 0 as the number of the first node. Then we filled in the "name" with the title of abstract where the sentence is extracted from. Third, we fill in the "content" with the tokenized sentence. Fourth, the "sent_num" represents the location of the sentence inside the abstract. In the end, the "label" for the evidence sentence is stored. We constructed the same structure for each node.

Transformer-XH on Evidence Graph

The evidence graph {X,E} is taken in by Transformer-XH to learn about verifying the claim to class y, with $y \in \{SUPPORTS, REFUTES, NOTENOUGHINFO\}$. First, the global representation of all text sequences is illustrated in Equation 3.15.

$$\mathcal{H}^L = TransformerXH(\mathcal{X}, \mathcal{E}) \tag{3.15}$$

Next, we attach the layer that aims at conducting the fact prediction onto the previous layer. This is done with [CLS] token of each node as depicted in Equation 3.16.



```
⊟ { } JSON
    ■ qid : 1
    ■ question : "0-dimensional biomaterials show inductive properties."
    ■ label : "NOT ENOUGH INFO"
  ⊟ [ ] node
    ⊟ { } 0
        ■ node_id : 0
        ■ name : "High-performance_neuroprosthetic_control_by_an_individual_with_tetraplegia."
      ⊟ [ ] context
          ■ 0 : "findings"
          ■ 1 : "the"
          ■ 2 : "participant"
          ■ 3 : "was"
          ■ 4 : "able"
          ■ 5 : "to"
          ■ 6 : "move"
          ■ 7 : "the"
          ■ 8 : "pro"
          ■ 9 : "##st"
          ■ 10 : "##hetic"
          ■ 11 : "limb"
          ■ 12 : "freely"
          ■ 13 : "in"
          ■ 14 : "the"
          ■ 15 : "three"
          ■ 16 : "-"
          ■ 17 : "dimensional"
          ■ 18 : "works"
          ■ 19 : "##pace"
          ■ 20 : "on"
          ■ 21 : "the"
          ■ 22 : "second"
          ■ 23 : "day"
          ■ 24 : "of"
          ■ 25 : "training"
          ■ 26 : "."
        ■ sent_num : 7
        ■ label : 0
    ⊞ { } 1
    ⊞ { } 2
    ⊞ { } 3
    ⊞ { } 4
    ⊞ { } 5
    [ ] edge
```

Figure 3.9: Evidence Graph for One Example Claim.

$$p(y|\tau) = softmax(Linear(\tilde{h}^{L}_{\tau,0})) \tag{3.16}$$

Later, we add the layer for measuring the importance of each node in the evidence graph as shown in Equation 3.17.

$$p(s|\tau) = softmax(Linear(\tilde{h}^{L}_{\tau,0})) \tag{3.17}$$

The final prediction is made with the the fact prediction of each node and their importance.

$$p(y|\mathcal{X}, \mathcal{E}) = \sum \tau p(s|\tau) \cdot p(y|\tau) \tag{3.18}$$

We perform label prediction in combination of the node prediction task and the claim verification task. The node prediction takes the cross entropy loss from Equation 3.16 and the evidence sentence label provided by SCIFACT as 1 for golden evidence and 0 for predicted evidence. The claim verification acquires the cross entropy loss from Equation 3.17 and golden label of the claim.

Chapter 4

Results and Discussion

In this chapter, we first discussed about experimental setups for conducting the experiments. It includes an introduction to the dataset and evaluation methods for each step of work. We then depicted the implementation details for the training. This contains the estimation of Transformer-XH model on the FEVER dataset, the training process for the label prediction step and the choice of parameters for the system. At last, the experimental results for each individual step are shown accompanied by evaluation. And the main focus is on the label prediction part.

4.1 Experimental Setups

4.1.1 Datasets

Our experiments are conducted on SCIFACT, the scientific fact verification benchmark by Wadden et al., 2020. Given a claim and a reliable corpus. i.e. Scientific articles, the task is to verify if the evidence in the corpus SUPPORTS, REFUTES or there is NOT ENOUGH INFO to verify the claim. The details for SCIFACT task is illustrated in Section 2.3. The training process is accompanied with the FEVER dataset, the details for that is depicted in Section 2.1.

4.1.2 Evaluation Methods

The evaluation methods for abstract retrieval and rationale selection is kept consistent with the work of Wadden et al., 2020. We compute the *HitOne* and *HitAll* accuracy for the abstract retrieval step. *HitOne* score represents that at least one abstract is retrieved correctly and *HitAll* score represents that all the abstracts are retrieved correctly. And then for the rationale selection step we compute the Precision, Recall and F1 score of the retrieved sentences. The Precision represents the percentage of number of relevant sentences retrieved divided by the total number of relevant sentences. The Recall represents the percentage of number of relevant sentences retrieved divided by the total number of sentences retrieved. F1 score find the average of Precision and Recall. Since our focus point on this paper is the label prediction step. We will be evaluating the classification results with not only the total label accuracy but also the label accuracy on different types of claims. The confusion matrix is also computed for the evaluation. The label accuracy is the percentage of the number of claims that has been classified correctly against the total number of claims which include all three labels. And the confusion matrix is a table layout to visualize the performance of our labeling method with row as the predicted classes and column as the true classes in our case.

4.2 Implementation Details

4.2.1 Evaluating Transformer-XH on FEVER

We first reproduced the experiments on FEVER dataset with Transformer-XH to verify the correctness of the previous work and confirm the usage of the model. The transformer-xh model is trained on FEVER dataset (around 145449 claims) for 2 epochs. The Label Accuracy on FEVER Dev dataset (9999 claims) is 78.05.

We reproduced the result with Label Accuracy of 78.03 in the evaluation process as we keep all parameters consistent with the previews work of Zhao et al., 2020 but changed maximum number of tokens from 130 to 128 due to the capability of computer memory. In Table 4.1 we show the result of Transformer-XH for Claim Verification on FEVER dataset from Zhao et al., 2020. The FEVER score is calculated following the work on document retrieval and evidence sentences selection of Z. Liu et al., 2019.

Dataset	Dev		Test	
Standard	Label Accuracy	FEVER	Label Accuracy	FEVER
Transformer-XH	78.05	74.98	72.39	69.07

Table 4.1: Transformer-XH for Claim Verification on FEVER Dataset

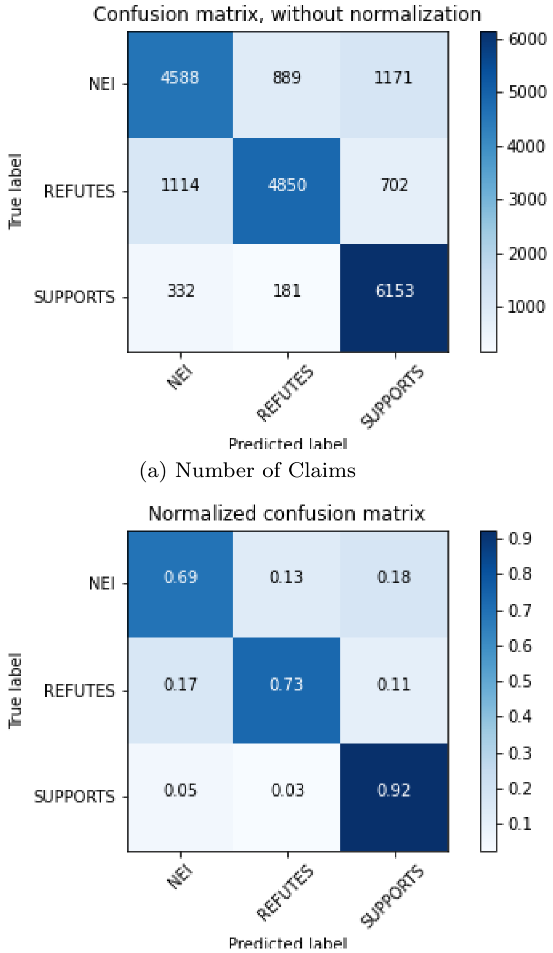
We generate the confusion matrix for label prediction result on the FEVER development set as depicted in Figure 4.1. As we can see from the confusion matrix, the prediction for claims with true label as SUPPORTS is with the highest accuracy of 0.92. Following with the claims with true label REFUTES and NOT ENOUGH INFO with the accuracy of 0.73 and 0.69.

We also tried to directly implement the model trained on FEVER to the SCIFACT development set. The labeling accuracy result is around 40% so we need to train the model on SCIFACT dataset for better performance.

4.2.2 Training the Label Prediction Step

We concatenate the training dataset of FEVER and SCIFACT into one evidence graph training set to perform the training on Transformer-XH. We also use the SCIFACT development set as evidence graph development set to evaluate the label prediction accuracy and adjust the model hyper parameters. After training for 4 epochs, the label accuracy convergence at 0.78. The loss gradually decreases and is stabilized at around 1900. The training process is visualized in Figure 4.2.

For the label prediction step, the model is trained with a single GeForce GTX 1080 GPU on the Linux system. It takes 315 minutes to train the model for one epoch on the FEVER+SCIFACT training set and we train the model for 4 epochs with 1260 minutes. The evaluation on SCIFACT



	NEI	REFUTES	SUPPORTS
NEI	4588	889	1171
REFUTES	1114	4850	702
SUPPORTS	332	181	6153

(a) Number of Claims

	NEI	REFUTES	SUPPORTS
NEI	0.69	0.13	0.18
REFUTES	0.17	0.73	0.11
SUPPORTS	0.05	0.03	0.92

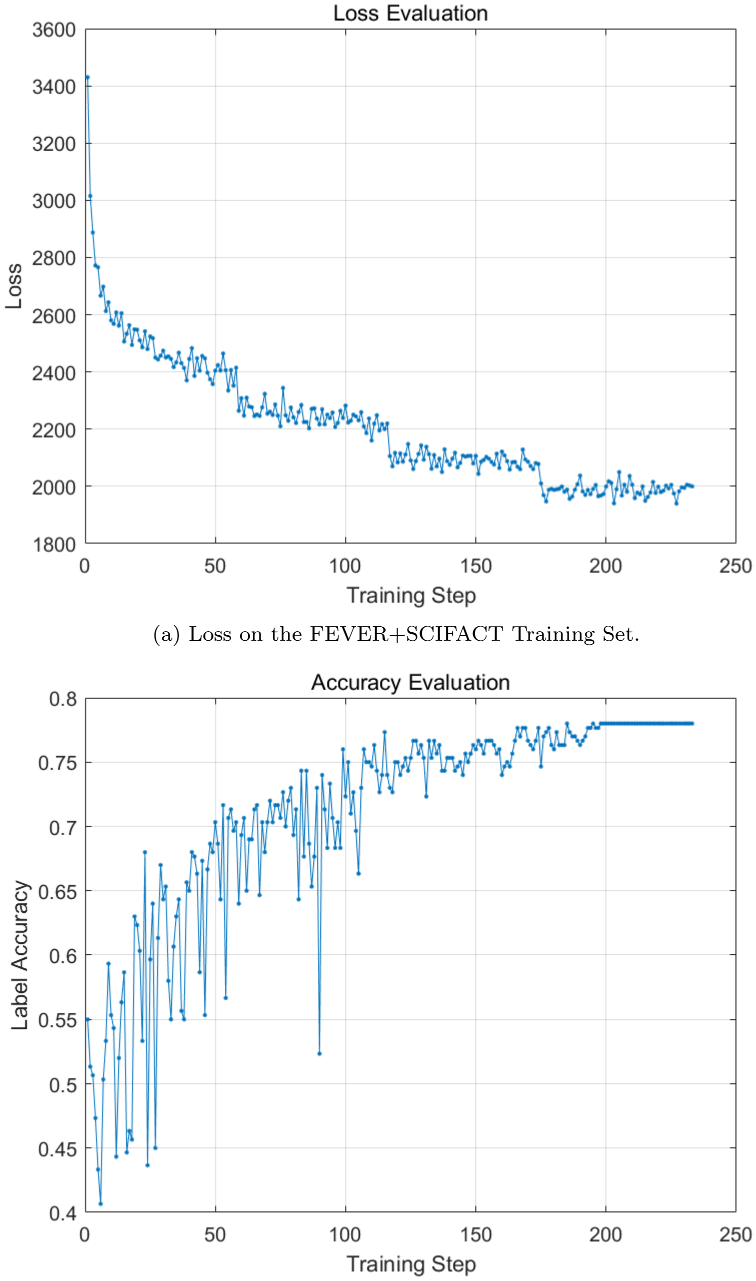
(b) Percentage of Claims

Figure 4.1: Confusion Matrix of FEVER Dev Set Label Prediction.

development set takes around 9 seconds each time.

4.2.3 Parameters for the System

We follow the experiment settings by the work of Wadden et al., 2020 for the abstract retrieval and rationale selection. The number of abstract retrieved for the abstract retrieval step is set to 3. The parameter is selected base on the results of SCIFACT development set. The parameters for the rationale selection step are described as follows. For the parameter $z_i = 1$ and $\hat{z}_i > t$ as illustrated in Section 2.4.1, the threshold t is selected as 0.5 for the training on SCIFACT dataset. When training on SCIFACT, the learning rate is settled to 1e-5 on the Transformer part and 1e-3 on the linear layer. The batch size is set to 256 and the best model is found with 20 epochs of training.



(a) Loss on the FEVER+SCIFACT Training Set.

(b) Label Accuracy on the SCIFACT Development Set.

Figure 4.2: Training Process of Transformer-XH on SCIFACT Task.

For the label prediction on SCIFACT task, we follow the previous implementation details of Zhao et al., 2020 on FEVER task. We initialize the Transformer-XH with the standard parameters of pre-trained BERT base model (Devlin et al., 2019). We set the parameters for extra hop attention arbitrarily and train directly. We implemented the three hop steps. We constructed the evidence graph as completely connected. When training on SCIFACT+FEVER training set, we put the learning rate to 1e-5 on the Transformer. The batch size is place to 1 and the model convergences after 4 epochs of training.

4.3 Results Evaluation

4.3.1 Abstract Retrieval

Although more evidence abstracts will be retrieved when we increase the k as illustrated in Table 4.2, we want to average between the precision and recall. Therefore, we take in the claim and corpus of the SCIFACT dataset and use TF-IDF to select the 3 most relevant abstracts. The unigrams and bigrams are extracted. In the end, the accuracy for at least one abstract is retrieved correctly is 0.8467 on the development set. The accuracy of retrieving all the abstracts correctly is 0.8333 on the development set.

Number of Abstracts	Hit One	Hit All
1	0.7467	0.7333
3	**0.8467**	**0.8333**
5	0.89	0.87
10	0.9267	0.91
20	0.95	0.94
50	0.9767	0.9633
100	0.9867	0.98

Table 4.2: Number of Abstracts and Hit Scores for Abstract Retrieval.

4.3.2 Rationale Selection

We take in the oracle experiments result from the abstract retrieval step, which means that the Hit One and Hit All score are both 1. Therefore we can evaluate the rationale selection step without the influence from the previous step. The result for the methods we are using (SCIBERT and RoBERTa-large) in comparison with other models is shown in Table 4.3.

SCIBERT

We reproduced the result of Wadden et al., 2020, fineturned the SCIBERT sentence encoder with SCIFACT dataset. We take golden evidence abstracts as input together with the claim and set the number of evidence sentences from each abstract to be flexible. The resulting Precision is 74.5, Recall is 74.3 and overall F1 score is 74.4.

RoBERTa-large

We reproduced the result of Wadden et al., 2020, fineturned the RoBERTa-large sentence encoder with SCIFACT dataset. We take golden evidence abstracts as input together with the claim and set the number of evidence sentences from each abstract to be flexible. The resulting Precision is 73.7, Recall is 70.5 and overall F1 score is 72.1.

		Rationale Selection		
Training Data	Model	P	R	F1
SCIFACT	BioMedRoBERTa	75.3	69.9	72.5
FEVER	RoBERTa-large	41.5	57.9	48.4
UKP Snopes	RoBERTa-large	42.5	62.3	50.5
FEVER+SCIFACT	RoBERTa-large	72.4	67.2	69.7
SCIFACT	RoBERTa-base	76.1	66.1	70.8
SCIFACT	RoBERTa-large	**73.7**	**70.5**	**72.1**
SCIFACT	SCIBERT	**74.5**	**74.3**	**74.4**

Table 4.3: Results for Rationale Selection.

4.3.3 Label Prediction

As illustrated in Section 4.2.2, the label accuracy of SCIFACT dev set convergence at 0.78. In this section, we dive into the details of the result. We first produce the confusion matrix for label prediction on the SCIFACT full dev dataset as illustrated in Figure 4.3. We can see that compared to the FEVER dataset as depicted in Figure 4.1, this time the prediction for claims with golden label as NOT ENOUGH INFO is with the highest accuracy of 0.96. Following by the claims with golden label as SUPPORTS with the accuracy of 0.87 and the claims with golden label as REFUTES with the accuracy of 0.30.

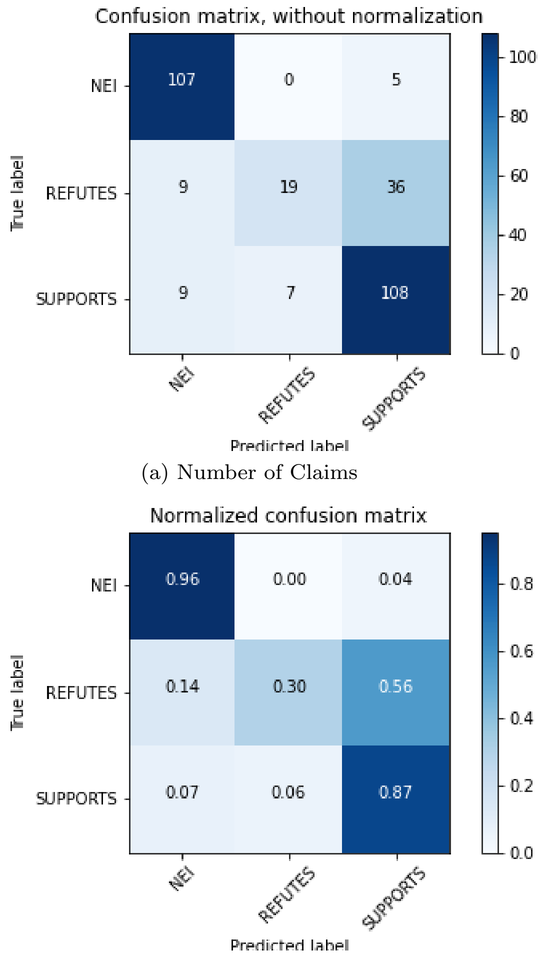
We anticipate that the low accuracy for claims with golden label as REFUTES could come with two reasons. First, there are only total 64 claims with golden REFUTES label and the bias could be large for even one wrongly labeled claim. Second, we constructed the evidence graph with only the golden evidence sentences for the claims with golden SUPPORTS and REFUTES labels. The amount of sentences for each claim is smaller than one for claims with golden NOT ENOUGH INFO labels. Therefore the textual entailment relationships between sentences and claim might be influenced.

We then studied the Label Accuracy on claims divided into Single Evidence and Multi Evidence categories. The Single Evidence represents that there's only one evidence sentence for the claim in the category. The Multi Evidence represents that there are multiple evidence sentences for the claim in the category. The result is shown in Table 4.4. We can observe that the Label Accuracy for Single Evidence claims 68.75 is lower than the overall Label Accuracy and the label Accuracy for Multi Evidence claims 82.35 is higher than the overall Label Accuracy. Also the Label Accuracy 82.35 for Multi Evidence claims exceeds the existing Label Accuracy results. We can conclude that the extra hop mechanism in Transformer-XH model brings a small step forward for the Label Prediction on the multi-evidence scientific claims labeling.

Model	Full	Single Evidence	Multi Evidence
Transformer-XH	78.0	68.75	82.35

Table 4.4: Single and Multi Evidence on SCIFACT Dev Claims.

The confusion matrix for label accuracy on the single evidence claims is depicted in Figure 4.4. As we have given each NOT ENOUGH INFO golden labeled claim 6 evidence sentences,



(a) Number of Claims

(b) Percentage of Claims

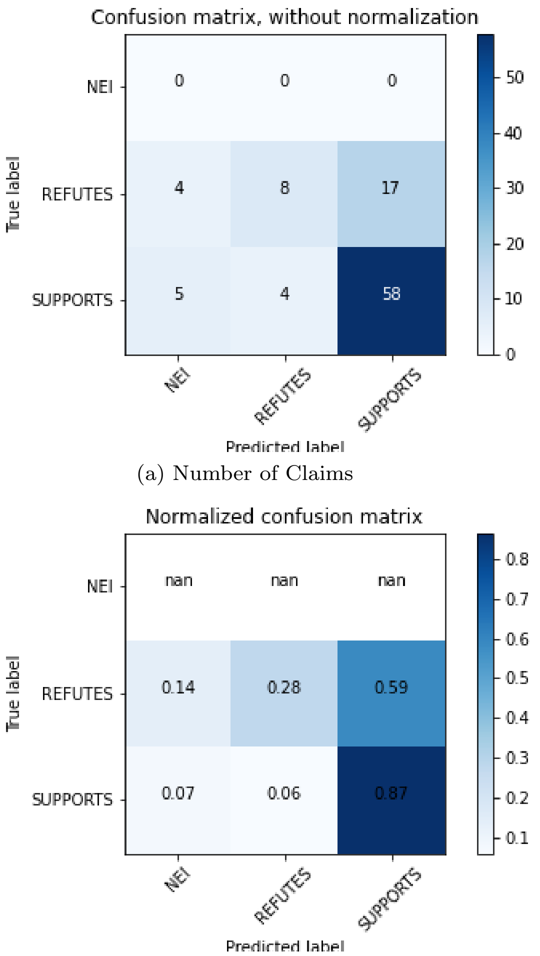
Figure 4.3: Confusion Matrix for Label Accuracy on the SCIFACT Development Set.

therefore there is no NOT ENOUGH INFO golden labeled claim in the single evidence claims set. The accuracy for SUPPORTS golden labeled claims is 0.87 and for REFUTES golden labeled claims is 0.28.

The confusion matrix for label accuracy on the multi evidence claims is depicted in Figure 4.5. The accuracy for NOT ENOUGH INFO golden labeled claims is 0.96, for SUPPORTS golden labeled claims is 0.88 and for REFUTES golden labeled claims is 0.31.

We can conclude that for both the SUPPORTS and REFUTES golden labeled claims, the model performs better on the multi evidence claims set. Also for both the FEVER and SCIFACT task, Transformer-XH performs better on successfully predict the SUPPORTS claims than the REFUTES claims.

In the end, we present the result of different models and training dataset for the label pre-



	NEI	REFUTES	SUPPORTS
NEI	0	0	0
REFUTES	4	8	17
SUPPORTS	5	4	58

(a) Number of Claims

	NEI	REFUTES	SUPPORTS
NEI	nan	nan	nan
REFUTES	0.14	0.28	0.59
SUPPORTS	0.07	0.06	0.87

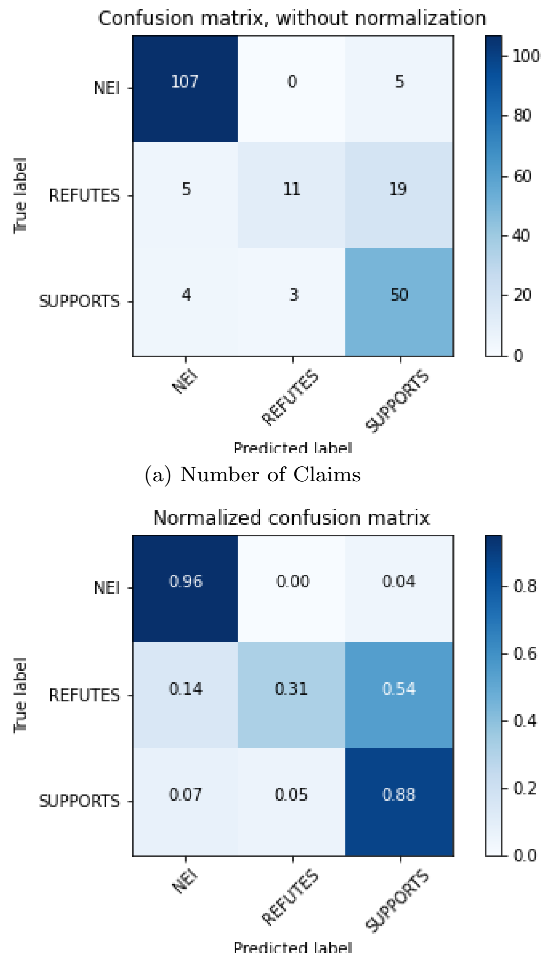
(b) Percentage of Claims

Figure 4.4: Confusion Matrix for Label Accuracy on the Single Evidence Claims.

diction step in Table 4.5. We can see that our model implemented gives the second place result on the full accuracy with 78.0 and exceeds the top result on Multi Evidence claims verification with 82.35.

4.3.4 Full Pipeline

In this paper we focus on the label prediction task. We exceed the label prediction accuracy on the multi-evidence claims. The full pipeline system will not be illustrated in our paper because the overall label accuracy achieves the state of art result but didn't exceeds the best score. And we performed the same methods of VERSCI as with the first and second step. Thus the result for the full pipeline is not at an optimized state. The state of art results for full pipeline system can be found in Table 2.10. We could try to implement the SciKGAT methods on the first and second step and get our optimized full pipeline result in the future work.



(a) Number of Claims

(b) Percentage of Claims

Figure 4.5: Confusion Matrix for Label Accuracy on the Multi Evidence Claims.

Training Data	Model	Label Prediction
SCIFACT	BioMedRoBERTa	71.7
SCIFACT	RoBERTa-base	62.9
SCIFACT	SCIBERT	69.2
SCIFACT	RoBERTa-large	75.7
UKP Snopes	RoBERTa-large	71.3
FEVER	RoBERTa-large	67.6
FEVER+SCIFACT	RoBERTa-large	**81.9**
FEVER+SCIFACT	Transformer-XH(Full)	**78.0**
FEVER+SCIFACT	Transformer-XH(Single Evidence)	68.75
FEVER+SCIFACT	Transformer-XH(Multi Evidence)	**82.35**

Table 4.5: Comparison on Different Training Dataset and Models for Label Prediction.

Chapter 5

Conclusion

In this chapter, we first give a conclusion of our progress and then discuss about the potential future work.

5.1 Existing Progress

We introduced the scientific Fact Extraction and VERification task in the Natural Language Processing domain of research. We investigated into the existing large-scale FEVER task and the neural architectures models implemented on it. And then we bring in the SCIFACT task and discuss about existing implementations and possible methods that could be conducted on the dataset. We also made a comparison between the FEVER and SCIFACT task.

Regarding the experiments, we followed the pipeline system and conducted experiments for abstract retrieval, rationale selection and label prediction individually. We use TF-IDF to extract k = 3 most relevant abstracts with the Hit One score 0.8467 and Hit All score 0.8333 as results. Then with the BERT-style model, we retrieved the evidence sentences from the golden abstracts. The best F1 score is 74.4 from model SCIBERT. At last,the Trasnformer-XH is implemented for label prediction with the evidence sentences. The full label accuracy is stated as 78.0 and is at the second place among results of other present methods. The Multi Evidence label accuracy is 82.35 which is better than the best score 81.9 from RoBERT-large.

5.2 Future Work

The rationale selection model could be trained with the continuous in-domain training method. And more models could be implemented for optimizing the second step result. We also could replace the method performed on abstract retrieval to either mention or key word based one, or the combination with TF-IDF. Regarding the label prediction part, we should investigate into the golden labeled REFUTES claims as the inaccuracy on the dev set mainly comes from this part. Also, the full pipeline evaluation should be conducted after optimizing the first two parts' result with state of art models.

Future work regarding scientific fact checking should not be limited to the collected scientific corpus. The study on verifying scientific claims with daily basis corpus like social media posts and normal website articles is a challenging task but can bring in huge application value.

Appendix A

Appendix A

A.1 Modeling Capabilities Explanation

- **Science background** includes knowledge of domain-specific lexical relationships (Wadden et al., 2020).
- **Directionality** requires understanding increases or decreases in scientific quantities (Wadden et al., 2020).
- **Numerical reasoning** involves interpreting numerical or statistical findings (Wadden et al., 2020).
- **Cause and effect** requires reasoning about counterfactual (Wadden et al., 2020).
- **Coreference** involves drawing conclusions using context stated outside of a rationale sentence (Wadden et al., 2020).

A.2 Source Articles by Journal

Journal	Count
BMJ	60
Blood	8
Cancer Cell	8
Cell	51
Cell Metabolism	10
Cell Stem Cell	41
Circulation	12
Immunity	33
JAMA	79
Molecular Cell	27
Molecular System Biology	5
Nature	29
Nature Cell Biology	26
Nature Communication	19
Nature Genetics	8
Nature Medicine	89
Nature Methods	1
Nucleic Acids Research	10
Plos Biology	36
Plos Medicine	38
Science	7
Science Translational Medicine	2
The Lancet	22
Other	120
Total	741

Table A.1: Corpus Source for SCIFACT Dataset (Wadden et al., 2020).

Bibliography

Alammar, J. (2018a). *The illustrated bert, elmo, and co. (how nlp cracked transfer learning).* http://jalammar.github.io/illustrated-bert/

Alammar, J. (2018b). *The illustrated transformer.* https://jalammar.github.io/illustrated-transformer/

Bekoulis, G., Papagiannopoulou, C., & Deligiannis, N. (2020). Fact extraction and verification – the fever case: An overview.

Beltagy, I., Lo, K., & Cohan, A. (2019). SciBERT: A pretrained language model for scientific text. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 3615–3620. https://doi.org/10.18653/v1/D19-1371

Chakrabarty, T., Alhindi, T., & Muresan, S. (2018). Robust document retrieval and individual evidence modeling for fact extraction and verification. *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*, 127–131. https://doi.org/10.18653/v1/W18-5521

Chen, D., Fisch, A., Weston, J., & Bordes, A. (2017). Reading wikipedia to answer open-domain questions.

Chen, Q., Zhu, X., Ling, Z.-H., Wei, S., Jiang, H., & Inkpen, D. (2017). Enhanced lstm for natural language inference. *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers).* https://doi.org/10.18653/v1/p17-1152

Dai, Z., Yang, Z., Yang, Y., Carbonell, J., Le, Q., & Salakhutdinov, R. (2019). Transformer-XL: Attentive language models beyond a fixed-length context. *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2978–2988. https://doi.org/10.18653/v1/P19-1285

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, *abs/1810.04805*. http://arxiv.org/abs/1810.04805

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding.

DeYoung, J., Jain, S., Rajani, N. F., Lehman, E., Xiong, C., Socher, R., & Wallace, B. C. (2020). ERASER: A benchmark to evaluate rationalized NLP models. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 4443–4458. https://doi.org/10.18653/v1/2020.acl-main.408

Fleiss, J. L. ((1971). Measuring nominal scale agreement among many raters. https://doi.org/10.1037/h0031619

Hanselowski, A., Zhang, H., Li, Z., Sorokin, D., Schiller, B., Schulz, C., & Gurevych, I. (2018). UKP-athene: Multi-sentence textual entailment for claim verification. *Proceedings of* *the First Workshop on Fact Extraction and VERification (FEVER)*, 103–108. https://doi.org/10.18653/v1/W18-5516

Hidey, C., & Diab, M. (2018). Team SWEEPer: Joint sentence extraction and fact checking with pointer networks. *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*, 150–155. https://doi.org/10.18653/v1/W18-5525

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., & Stoyanov, V. (2019). Roberta: A robustly optimized BERT pretraining approach. *CoRR*, *abs/1907.11692*. http://arxiv.org/abs/1907.11692

Liu, Z., Xiong, C., Dai, Z., Sun, S., Sun, M., & Liu, Z. (2020). Adapting open domain fact extraction and verification to COVID-FACT through in-domain language modeling. *Findings of the Association for Computational Linguistics: EMNLP 2020*, 2395–2400. https://doi.org/10.18653/v1/2020.findings-emnlp.216

Liu, Z., Xiong, C., & Sun, M. (2019). Kernel graph attention network for fact verification. *CoRR*, *abs/1910.09796*. http://arxiv.org/abs/1910.09796

Liu, Z., Xiong, C., Sun, M., & Liu, Z. (2020). Fine-grained fact verification with kernel graph attention network. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 7342–7351. https://doi.org/10.18653/v1/2020.acl-main.655

Lo, K., Wang, L. L., Neumann, M., Kinney, R., & Weld, D. (2020). S2ORC: The semantic scholar open research corpus. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 4969–4983. https://doi.org/10.18653/v1/2020.acl-main.447

Luken, J., Jiang, N., & de Marneffe, M.-C. (2018). QED: A fact verification system for the FEVER shared task. *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*, 156–160. https://doi.org/10.18653/v1/W18-5526

Malon, C. (2018). Team papelo: Transformer networks at FEVER. *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*, 109–113. https://doi.org/10.18653/v1/W18-5517

Manning, C., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S., & McClosky, D. (2014). The Stanford CoreNLP natural language processing toolkit. *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, 55–60. https://doi.org/10.3115/v1/P14-5010

Nie, Y., Chen, H., & Bansal, M. (2018a). Combining fact extraction and verification with neural semantic matching networks.

Nie, Y., Chen, H., & Bansal, M. (2018b). Combining fact extraction and verification with neural semantic matching networks. *CoRR*, *abs/1811.07039*. http://arxiv.org/abs/1811.07039

Nie, Y., Chen, H., & Bansal, M. (2019). Combining fact extraction and verification with neural semantic matching networks. *Proceedings of the AAAI Conference on Artificial Intelligence*, *33*, 6859–6866. https://doi.org/10.1609/aaai.v33i01.33016859

Nie, Y., Wang, S., & Bansal, M. (2019). Revealing the importance of semantic retrieval for machine reading at scale. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2553–2566. https://doi.org/10.18653/v1/D19-1258

Parikh, A. P., Täckström, O., Das, D., & Uszkoreit, J. (2016). A decomposable attention model for natural language inference. *EMNLP*.

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L. (2018). Deep contextualized word representations. *CoRR*, *abs/1802.05365*. http://arxiv.org/abs/1802.05365

Portelli, B., Zhao, J., Schuster, T., Serra, G., & Santus, E. (2020). Distilling the evidence to augment fact verification models. *Proceedings of the Third Workshop on Fact Extraction and VERification (FEVER)*, 47–51. https://doi.org/10.18653/v1/2020.fever-1.7

Riedel, B., Augenstein, I., Spithourakis, G. P., & Riedel, S. (2017). A simple but tough-to-beat baseline for the fake news challenge stance detection task. *ArXiv*, *abs/1707.03264*.

Schroff, F., Kalenichenko, D., & Philbin, J. (2015). Facenet: A unified embedding for face recognition and clustering. *CoRR*, *abs/1503.03832*. http://arxiv.org/abs/1503.03832

Soleimani, A., Monz, C., & Worring, M. (2019). BERT for evidence retrieval and claim verification. *CoRR*, *abs/1910.02655*. http://arxiv.org/abs/1910.02655

Soleimani, A., Monz, C., & Worring, M. (2020). Bert for evidence retrieval and claim verification. In J. M. Jose, E. Yilmaz, J. Magalhães, P. Castells, N. Ferro, M. J. Silva, & F. Martins (Eds.), *Advances in information retrieval* (pp. 359–366). Springer International Publishing.

Stammbach, D., & Ash, E. (2020). E-fever: Explanations and summaries forautomated fact checking. In E. D. Cristofaro & P. Nakov (Eds.), *Proceedings of the 2020 truth and trust online conference (TTO 2020), virtual, october 15-17, 2020* (pp. 32–43). Hacks Hackers. https://truthandtrustonline.com/wp-content/uploads/2020/10/TTO04.pdf

Stammbach, D., & Neumann, G. (2019). Team DOMLIN: Exploiting evidence enhancement for the FEVER shared task. *Proceedings of the Second Workshop on Fact Extraction and VERification (FEVER)*, 105–109. https://doi.org/10.18653/v1/D19-6616

Taniguchi, M., Taniguchi, T., Takahashi, T., Miura, Y., & Ohkuma, T. (2018). Integrating entity linking and evidence ranking for fact extraction and verification. *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*, 124–126. https://doi.org/10.18653/v1/W18-5520

Thorne, J., Vlachos, A., Christodoulopoulos, C., & Mittal, A. (2018). FEVER: A large-scale dataset for fact extraction and verification. *NAACL-HLT*.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention is all you need. *CoRR*, *abs/1706.03762*. http://arxiv.org/abs/1706.03762

Wadden, D., Lo, K., Wang, L. L., Lin, S., van Zuylen, M., Cohan, A., & Hajishirzi, H. (2020). Fact or fiction: Verifying scientific claims. *arXiv preprint arXiv:2004.14974*.

Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., Klingner, J., Shah, A., Johnson, M., Liu, X., Kaiser, L., Gouws, S., Kato, Y., Kudo, T., Kazawa, H., ... Dean, J. (2016). Google's neural machine translation system: Bridging the gap between human and machine translation. *CoRR*, *abs/1609.08144*. http://arxiv.org/abs/1609.08144

Yang, Z., Dai, Z., Yang, Y., Carbonell, J. G., Salakhutdinov, R., & Le, Q. V. (2019). Xlnet: Generalized autoregressive pretraining for language understanding. *CoRR*, *abs/1906.08237*. http://arxiv.org/abs/1906.08237

Yin, W., & Schütze, H. (2018). Attentive convolution: Equipping CNNs with RNN-style attention mechanisms. *Transactions of the Association for Computational Linguistics*, *6*, 687–702. https://doi.org/10.1162/tacl_a_00249

Yoneda, T., Mitchell, J., Welbl, J., Stenetorp, P., & Riedel, S. (2018). UCL machine reading group: Four factor framework for fact finding (HexaF). *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*, 97–102. https://doi.org/10.18653/v1/W18-5515

Zhao, C., Xiong, C., Rosset, C., Song, X., Bennett, P., & Tiwary, S. (2020). Transformer-xh: Multi-evidence reasoning with extra hop attention. *International Conference on Learning Representations*. https://openreview.net/forum?id=r1eIiCNYwS

Zhong, W., Xu, J., Tang, D., Xu, Z., Duan, N., Zhou, M., Wang, J., & Yin, J. (2019). Reasoning over semantic-level graph for fact checking. *CoRR*, *abs/1909.03745*. http://arxiv.org/abs/1909.03745

Zhong, W., Xu, J., Tang, D., Xu, Z., Duan, N., Zhou, M., Wang, J., & Yin, J. (2020). Reasoning over semantic-level graph for fact checking.

Zhou, J., Han, X., Yang, C., Liu, Z., Wang, L., Li, C., & Sun, M. (2019). GEAR: graph-based evidence aggregating and reasoning for fact verification. *CoRR*, *abs/1908.01843*. http://arxiv.org/abs/1908.01843