QATAR UNIVERSITY

COLLEGE OF ENGINEERING

DID I SEE IT BEFORE? RETRIEVING PREVIOUSLY CHECKED CLAIMS OVER TWITTER

BY

WATHEQ AHMAD MANSOUR

A Thesis Submitted to

the College of Engineering

in Partial Fulfillment of the Requirements for the Degree of

Master of Science in Computing

January 2022

© 2022. WATHEQ AHMAD MANSOUR. All Rights Reserved.

COMMITTEE PAGE

The members of the Committee approve the Thesis of
WATHEQ AHMAD MANSOUR defended on 25/11/2021.

Dr. Abdulaziz Al-Ali
Thesis Supervisor

Dr. Tamer Elsayed
Thesis Co-Supervisor

Dr. Preslav Nakov
Committee Member

Dr. Junaid Qadir
Committee Member

Dr. Abdelaziz Bouras
Committee Member

Approved:

Khalid Kamal Naji, Dean, College of Engineering

ABSTRACT

MANSOUR, WATHEQ, A., Masters : January: 2022, Master of Science in Computing

Title: Did I See it Before? Retrieving Previously Checked Claims over Twitter

Supervisors of Thesis: Dr. Abdulaziz Al-Ali and Dr. Tamer Elsayed.

With the proliferation of fake news in the last few years, especially during COVID-19, combating the spread of misinformation has become a social and political urgent need. Fact-checkers and journalists need to identify claims that were previously verified by a reputable fact-checking organization before inspecting the claim veracity. Many claims showed up repeatedly but at different time periods and different forms. In this thesis, we propose an *automated* approach to retrieve claims that have been already *manually*-verified by professional fact-checkers. Our proposed approach uses recent powerful BERT (BERT is a Transformer-based machine learning technique that can be used to address several Natural Language Processing problems effectively) variants as rerankers in monoBERT fashion. MonoBERT is a point-wise ranking approach that uses a BERT-based model to assign a relevance score for query-document pair. Additionally, we study the impact of using different fields of the verified claim during training and inference phases. Experimental results show that our proposed pipeline outperforms the state-of-the-art approaches on two public English datasets and one Arabic dataset by a remarkable margin. Moreover, we are the first to develop a system for the Arabic language.

DEDICATION

To my dear mother, father, and family who have been showering me with love and support throughout my whole life.

To my beloved wife without whom I could not have overcome the hard challenges during the master journey.

ACKNOWLEDGMENTS

Thank words are not enough for my supervisors, Dr. Abdulaziz Al-Ali and Dr. Tamer Elsayed. I really appreciate their efforts toward the completion of the thesis. Without their great guidance, encouragement, and patience, I could not have done this work.

I acknowledge Fatima Haouari for her valuable contribution toward the creation of the claim retrieval dataset. I would like to thank Zein Sheikhali for her efforts in creating the AraFacts dataset, which was the seed for creating the claim retrieval dataset.

This thesis was made possible by NPRP grant No.: NPRP11S-1204-170060 from the Qatar National Research Fund (a member of Qatar Foundation). The statements made herein are solely the responsibility of the authors.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1: INTRODUCTION

The massive spread of misinformation has a negative impact on many governments, public figures, and organizations, among others [1]. Moreover, economic status and financial markets are affected substantially by fake news. Fake news might distort and manipulate public opinions, which might lead to negative effects on election results. Due to the availability and widespread of social media platforms, users could use online social networks (OSN) to spread fake news to millions of people in just a few seconds or minutes. Nowadays, most people depend on social media platforms to get the latest news as these platforms provide easy access compared to traditional news agencies. During the 2010 Dutch and British elections, Broersma and Graham [2] showed that about half of the Dutch and nearly a quarter of British candidates posted their thoughts and visions over Twitter, as Figure 1.1 shows that for the 2010 British Elections. Consequently, most of these tweets were discussed in news articles.

Party	No. of twittering candidates	No. of candidates	Twittering candidates (%)
Liberal Democrats	168	631	27
Labour	141	631	22
Conservative	122	631	19
Total	431	1893	23

Figure 1.1: Number of twitter candidates per Party for the 2010 UK General Election [2]

That created an urgent social need to combat the spread of misinformation. As a response, many fact-checking organizations, e.g., Politifact[1] and FullFact[2], arose in the last few years. However, most of these organizations perform fact-checking *manually*, which is indeed very time-consuming. The huge amount of misinformation being spread

[1] https://www.politifact.com/
[2] http://fullfact.org/

over social media surpasses the human ability to fact-check them manually.

To address those issues, several research directions were pursued to develop automated systems that identify check-worthy claims and investigate their factuality [3]–[5]. It has been shown in the literature that retrieving previously fact-checked claims could be an integral part of an end-to-end automated fact-checking system [6][7]. Figure 1.2 shows general steps of such a system [6].

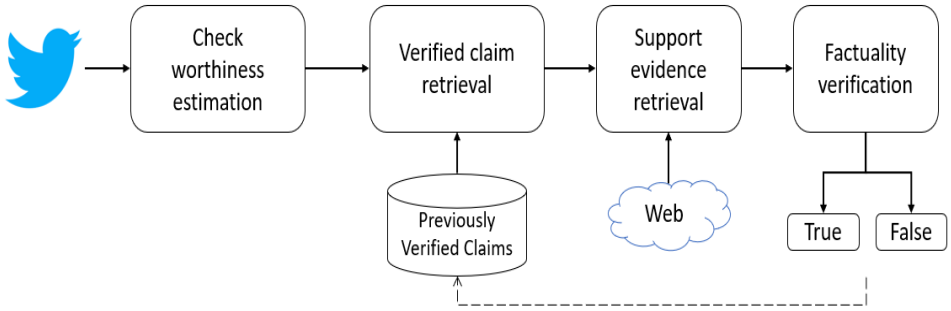

Figure 1.2: Steps of an automated end-to-end fact-checking system [6].

Indeed, many viral claims show up repeatedly at different time periods, such as those appearing during the COVID-19 period. Recognizing that those claims have been already verified by professional fact-checkers has several advantages. It helps mitigate the negative effect of spreading fake news on both society and individuals. Furthermore, it helps journalists put their interviewees on the spot in real-time whenever incorrect facts are stated. It also allows more time for the automated and manual verification systems to focus on verifying unchecked claims.

To that end, in this thesis, we tackle the problem of *claim retrieval over Twitter*, defined as follows: given a tweet that includes a claim (denoted as the query) and a collection of previously checked claims, we aim to retrieve all relevant previously-verified claims with respect to the input tweet. We frame the problem as a *ranking* problem over

a collection of previously-verified claims, where, for a given containing-claim tweet, we rank the most relevant previously-verified claims at the top. Figure 1.3 illustrates an example tweet and corresponding verified claim. The problem is challenging from two aspects. First, tweets' text is typically informal and lacks context due to the length limitations. Second, claims can be phrased in different forms, calling for semantic matching.

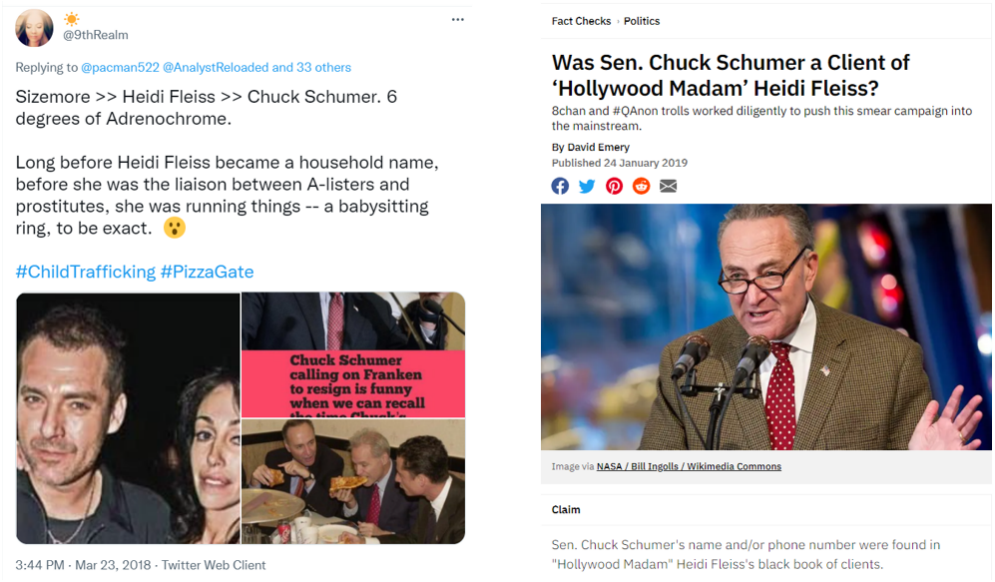

Figure 1.3: Example of the claim retrieval problem: tweet (left) and verified claim (right).

To address the problem, we propose a three-step pipeline. Tweet preprocessing is performed first, where it is expanded by extracting information from embedded URLs, images, and videos. A recall-oriented retrieval step follows, where a classical light-weight retrieval model is used to retrieve an initial set of potentially relevant claims. Finally, the set is re-ranked by a precision-oriented neural network-based model. Several approaches exist for each of these steps. We conducted our experiments on three datasets that are commonly used in the literature; two are in English, namely CheckThat 2020 English (CT2020-En) [8], CheckThat 2021 English (CT2021-En) [9], and a third in

Arabic, CheckThat 2021 Arabic (CT2021-Ar) [9].

Our contributions are seven-fold:

- Our proposed approach outperforms the state-of-the-art approaches for the English language on two datasets.
- We are the first to build an Arabic dataset for the claim retrieval task.
- We are the first to develop a claim retrieval system for the Arabic language.
- We compare several point-wise BERT-based learning-to-rank techniques.
- We examine the impact of using different fields of a given verified claim on the training phase and during inference.
- We study the effectiveness of utilizing duoBERT model for the claim retrieval task.
- We study the impact of integrating the curriculum learning technique during training of point-wise rerankers.

Throughout this thesis, we aim to answer the following research questions:

RQ1 What is the impact of the preprocessing steps on the performance of the initial retrieval stage? (Section 5.1)

RQ2 Does a monoBERT reranker improve the performance over the initial retrieval stage? What is the best BERT variant for the task? (Sections 5.2)

RQ3 How can we effectively leverage the title and the description of verified claims in training and inference? (Section 5.3)

RQ4 Does applying curriculum learning in claim retrieval setup enhance the performance? (Section 5.4)

RQ5 Does duoBERT technique provide an improvement over the point-wise approaches? (Section 5.5)

RQ6 What will the performance of the proposed approach be on Arabic data? (Section 5.6)

The remainder of this thesis is organized as follows. Chapter 2 discusses related work. Chapter 3 presents our proposed methodology. Chapter 4 illustrates the experimental setup. Chapter 5 details our experimental evaluation and results. Chapter 6 provides concluding remarks and suggests future directions.

CHAPTER 2: RELATED WORK

2.1. Ranked Retrieval with BERT

In information retrieval, document ranking is a well-known problem that is defined as follows: given a query (representing an information need of a user) and a collection of documents, the system goal is to rank the relevant documents at the top in order to maximize some metric, e.g., mean average precision (MAP). The claim retrieval can be considered as a document ranking problem where tweets (the claim within a tweet) are considered queries, and the documents are represented by the previously fact-checked claims.

The advent of deep learning has boosted the performance of the retrieval models for the better. BERT (Bidirectional Encoder Representations from Transformers) is a recent powerful deep learning technique that was proposed by Devlin et al. [10]. BERT can be used to address several Natural Language Processing (NLP) problems effectively. BERT helps in understanding the context within a sentence. Many extensions and modifications were applied to this model afterward.

Due to its high effectiveness, BERT has been used extensively in the literature as a ranking model. In such context, for a given pair of query and document, BERT can predict the relevance score between the query and document (generally the score between 0 and 1). Then, this score is compared with scores of other documents to decide the rank of this document.

2.2. Claim Retrieval

The claim retrieval problem has garnered much attention from the industry and academia. From an industrial perspective, Google launched its Fact Check Explorer tool [11] that enables users to search through trusted fact-checking websites for a specific topic or claim. However, this tool cannot address complex claims and has poor performance when used with Arabic text.

There are two lines of research that target the claim retrieval problem. The first line is concerned with matching a given query with the body of a fact-checking *article* like [12]–[14]. Shaar et al. [12] used BM25 for initial ranking. Then, they utilized the scores of BM25 and sentence-BERT for training a RankSVM reranker. Vo and Lee [13] suggested a framework that uses both texts and images to search for fact-checking articles achieving around 5% improvement over nine state-of-the-art neural ranking baselines. Sheng et al. [14] proposed a transformer-based reranker that captures the key sentences within a fact-checked article, then exploits them to estimate the relevance score. Although the system proposed by Sheng et al. al. [14] outperformed [12], [13], it is more complex and requires multiple steps to identify the key sentences.

The second line aims to link the given query with previously fact-checked *claims*. The Verified Claim Retrieval shared task in CheckThat Lab 2020 and 2021 [8][15] is a clear representative of this type of research. The goal is to detect whether a claim-containing tweet was previously checked with respect to a collection of verified claims. Our focus is directed toward this category of solutions as they are self-explainable and could be used in real-life scenarios. While the task was proposed in the English language only in 2020 [8], it was presented in two languages in 2021 [9], [15].

Retrieving already verified claims for a stated claim in political debates is another variation of this task which was proposed first by Shaar et al. [16]. To capture the context of the claims made in a debate, Shaar et al. [16] used Transformer-XH to model the local and global contexts. They found that modeling the context of the debate is more important than modeling the context of a fact-checking article.

Multiple teams participated in the shared task of the CheckThat! lab 2020 [8] and 2021 [9], [15] and followed different strategies in preprocessing and ranking. Bouziane et al. [17], the top team in CheckThat! lab 2020, utilized multi-modal data and augmented the data with a similar dataset. After that, they fine-tuned the pre-trained RoBERTa model with adversarial hard negative examples to rerank the tweet-verified-claim pairs. We adopt this approach as a strong baseline to compare against for the 2020 dataset. RoBERTa [18] is a variation of BERT.

Passaro et al. [19] adopted a two-phase strategy. In the first phase, they used sentence-BERT [20] to produce a high cosine similarity score to gold pairs. In the second phase, they fine-tuned a sentence-BERT model to perform the classification task in which the model gives one as an output if the pair constitutes a correct match and zero otherwise. Finally, they rerank the pairs based on the classification score.

Thuma et al. [21] first retrieved the top-1000 relevant tweet-claim pairs for each tweet using DPH information retrieval weighting model. Then, for each retrieved document, they extracted some query-dependent features. Then, they deployed LambdaMart, a learning to rank technique, on the feature vector of top-k documents. McDonald et al. [22] chose the features to be a combination of scores of TF-IDF, BM25, and cosine similarity, as done in [12]. After that, they used the extracted features in training multiple machine learning models such as Logistic Regression, linear regression, and Linear

SVM. The sum of multiplying each extracted feature with a corresponding coefficient is considered as the reranking score.

Chernyavskiy et al. [23], the winning team in the CheckThat! lab 2021, made some modifications to the system proposed by Shaar et al. [12]. They used the scores of TF-IDF and fine-tuned sentence-BERT as a feature vector to train a LambdaMART reranker. Their system outperformed the other teams and the organizers' baseline by a large margin. We select this system as a competitor baseline for the 2021 dataset.

Recently, Kazemi et al. [24] created a dataset of 2343 pairs in five languages (English, Hindi, Bengali, Malayalam, and Tamil) for claim matching. Their work mainly focused on identifying the matching between either pairs of verified claims "only" or pairs of social media content "only" (namely, WhatsApp messages). However, only 7% of their dataset are pairs of social media content and fact-checked claims.

Most of these methods did not consider the importance of extracting the information from URLs within a tweet (such as getting the title of a web page article or a short description of an image or video), replacing the Twitter user handles with their usernames, or determining which BERT variant is the most suitable for this kind of task.

2.3. Curriculum Learning

Generally, *Continuation Methods* is adopted if the objective function is non-convex, and applying direct optimization over it could lead to stuck in a poor local minimum [25]. Curriculum learning (CL) is considered as a particular case of *Continuation Methods* that can address the previously mentioned problem [26]. The main idea of curriculum learning is to organize the training process in a path where the easiest training examples are presented first to the learner, then the complexity of the training examples

is increased gradually. The rationale of this strategy is to make the learner employs easy training examples in understanding more complicated ones. CL has shown its effectiveness for training neural networks in multiple domains like NLP [27][28], image representation [29], open domain answer reranking [30]. To the best of our knowledge, there is no prior work that exploited CL in the claim retrieval task.

CHAPTER 3: THE PROPOSED PIPELINE

Our proposed approach is a simple three-step pipeline: preprocessing, initial retrieval, and reranking. In this section, we present each of them in detail. The proposed pipeline is illustrated in Figure 3.1.

3.1. Step 1: Preprocessing

Tweets are typically short and usually contain components that are references rather than textual content, such as URLs, user mentions, images, or videos, which might be noisy for our task. To add more context to the tweets, we have expanded those components. We converted URL links to their corresponding web page titles, replaced the mentions with their corresponding user names, obtained the title for the embedded images using reverse image search similar to [17], and added a short description of the embedded videos using reverse image search for video. As illustrated in Figure 3.2, more contextual information is added to the tweet after *expanding* the user mention and the embedded image. Generally, usernames represent the main objects within a tweet. As a result, eliminating such information could lead to a decrease in retrieval performance.

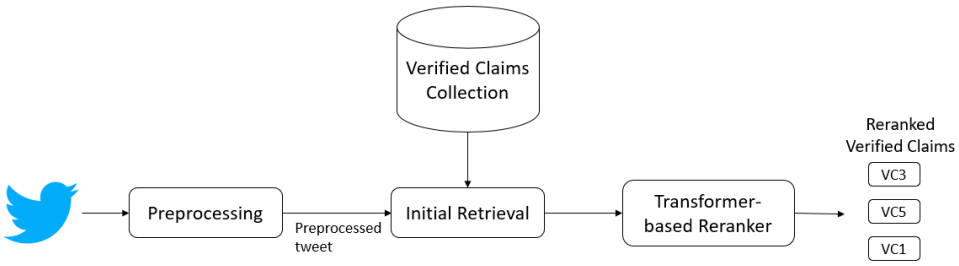

Figure 3.1: Illustration of the proposed methodology.

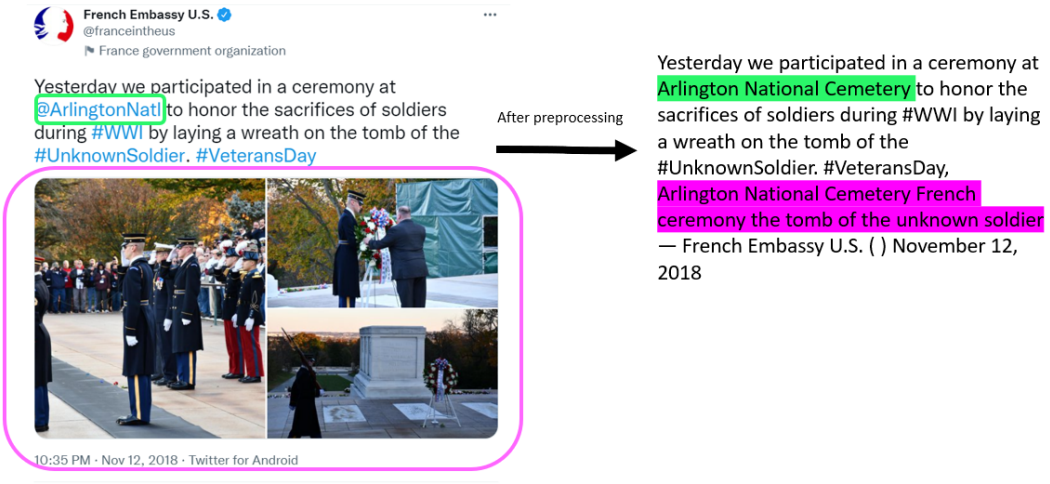

Figure 3.2: Example of a tweet before and after preprocessing. Notice the expanded user mention and the embedded image.

3.2. Step 2: Initial Retrieval

The second step in the pipeline retrieves an initial short list of potentially relevant claims. The goal is to retrieve as many as possible of them, i.e., maximize the recall in the cheapest way, to prepare for the reranking step that aims to push the relevant ones to the top of the list. To achieve that, we leverage the classical retrieval models that are lexical-based, i.e., relying on term overlap, e.g., BM25 [31].

3.3. Step 3: Reranking using Point-wise Approaches

The last step of the pipeline aims to improve the effectiveness of the initial retrieved list, using a more expensive reranker that is applied to the initial short list. Since transformer-based models (e.g., BERT [10]) have shown great success in the information retrieval field, we chose to use BERT-variants models, e.g., monoBERT [32]. The model takes a query (the tweet in our context) and a document (a retrieved verified claim) and classifies the claim based on its relevance with respect to the tweet, using a classification

layer on top of the neural architecture. The relevance score provided by the classifier is eventually used to rerank the retrieved claims. This is an example of point-wise learning-to-rank models.

The use of BERT as a reranker is not new. However, the novelty in our contribution is mainly characterized by the way we build the training set and the strategy of training and testing the reranker. We choose the negative pairs in the training set so that they cover what the model will be exposed to at inference time. Additionally, for each verified claim in the used datasets, there are two fields. The first one is "VClaim", which is a reformulated text version of the original claim. The second one is "Title", which is the title of the article that fact-checks the claim and represents a summary of the claim. We opt to train and to test using different combinations of "VClaim" and "Title" fields aiming to study the effect of each field and to improve the retrieval effectiveness.

3.4. Reranking using DuoBERT Technique

Toward reaching the best performance, we opt to apply duoBERT [32] technique within the claim retrieval framework. Basically, duoBERT approach accepts the query and a pair of documents as input and produces a score that determines the extent to which the first document is more relevant than the second document with respect to the given query. Figure 3.3 shows the required steps before performing duoBERT and the relation between monoBERT and duoBERT rerankers. MonoBERT reranks a subset of the initial list retrieved by BM25 (an example of a lightweight lexical retrieval model) by selecting only the top-k verified claims. Then, the reranked list is fed into duoBERT, which in turn, utilizes them to form triplets (VC_i, VC_j, q). Finally, duoBERT predicts the relevance score for each triplet and reranks them accordingly.

Here, a score of 1 means the first document is more relevant than the second, and 0 refers to the opposite. Since the complexity of duoBERT is quadratic, the size of the list to-be ranked should be chosen wisely to avoid costly execution time. We follow Nogueira et al. [32] by selecting the list produced by monoBERT reranker as an input for the duoBERT reranker implying a four-step process.

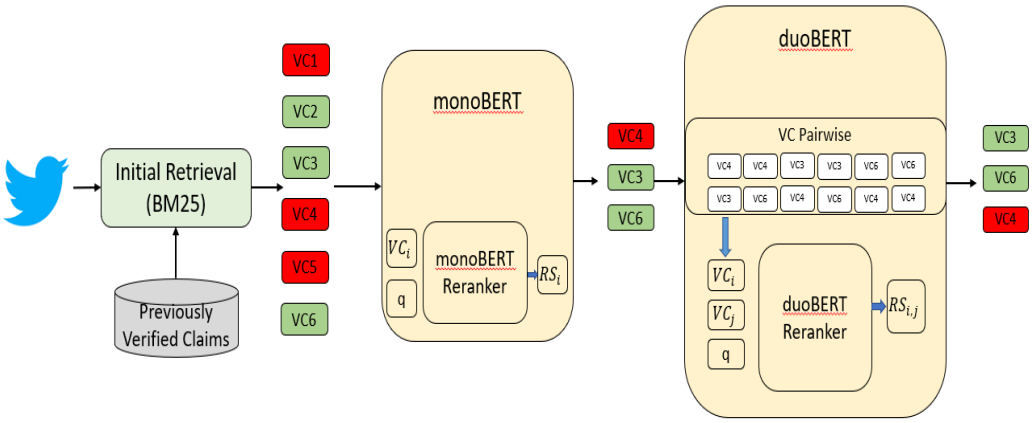

Figure 3.3: Illustration of ranking using monoBERT and duoBERT techniques in a multi-stage reranking manner. First, BM25 retrieves a list of verified claims (VCs). Next stage, monoBERT predicts the relevance (reranking) score for each (VC_i, q) pair. Finally, duoBERT forms triplets (VC_i, VC_j, q) from the previous stage, predicts the relevance score for each triplet and reranks them accordingly [32].

3.5. Curriculum Learning for Claim Retrieval

In a normal fine-tuning setup, model training treats all training examples equally. However, in a claim retrieval task, a training set contains easy and hard training examples (query-document pairs). Specifically, if the document in the training pair is relevant to the query and gets a high rank in the initial retrieval, then it is considered as an **easy** training example. Similarly, the low rank of a non-relevant document is considered easy as well. However, suppose the retrieval algorithm gives a high rank to a non-relevant document or a low rank to a relevant document. Such pairs (query-document) are

considered **hard** examples.

To emphasize these harder examples during training, we choose to employ curriculum learning (CL) technique [26]. We employ CL technique in the claim retrieval task by assigning different weights to the training examples through a heuristic. Initially, we assign high weights to the easy training examples while assigning low weights to the hard examples. Then, we increase the hard pairs' weights progressively to smooth the imbalance resulted from assigning higher weights from some examples over others. Eventually, all training examples will end up having the same weight. This is, in fact, similar to the approach by MacAvaney et al. [30] where a difficulty function is defined as hardness representative for a training sample.

MacAvaney et al. [30] used BM25 to design three difficulty functions. However, BM25 captures only the lexical hardness of a training sample. Therefore, we propose a novel difficulty function that captures both the contextual and lexical hardness for a training sample. The proposed difficulty function is a harmonic mean of two values. The first one is the BM25 reciprocal rank of the training sample, and the second one is the relevance score predicted by monoBERT reranker.

Formally, for a given query *q* and a verified claim *vc* in a collection of verified claims *C*, let k be the rank of *vc* retrieved by BM25. Then, $recip_q(vc) = \frac{1}{k}$ refers to the BM25 reciprocal rank of a verified claim *vc* for a given query *q* over the collection *C*. We omit *C* for simplicity. We denote the monoBERT relevance score between a query *q* and a verified claim *vc* with $RS(q, vc)$ where the values of both $recip_q(vc)$ and $RS(q, vc)$ are between 0 and 1. To that end, we propose our novel difficulty function as the harmonic mean between $recip_q(vc)$ and $RS(q, vc)$. We choose the harmonic mean function as it gives the least mean compared to the arithmetic and geometric means. As a result, the

verified claim will get a higher penalty when it goes down in the ranked list, i.e., it will be considered a much harder example. The formula of the proposed difficulty function is defined as follows:

$$D(q, vc) = \frac{2 * recip_q(vc) * RS(q, vc)}{recip_q(vc) + RS(q, vc)} \tag{3.1}$$

While the proposed function assigns a value near to one for easy training examples, it assigns a value close to zero for hard ones. To integrate the difficulty function during the training phase, we apply the same weighting function proposed by MacAvaney et al. [30]. The weighting function gives a weight for the difficulty value of a training sample based on the current epoch. The rationale of using a weighting function is to gradually make all training samples have equal weights. In other words, at epoch *m*, all training samples will end up having a weight of one. We refer to *m* as the curriculum threshold, which is a hyperparameter that we tune in our experiments.

3.6. Building Arabic Dataset for Claim Retrieval

We contribute to the creation of the first Arabic dataset for claim retrieval task [9]. Benefiting from the AraFacts [33] dataset, we select a subset of 1,274 verified claims so that the fact-checking article of these claims contains at least one stated tweet. Next, we build the gold tweet-verified-claims pairs by sticking to the following guidelines:

1. Select Arabic tweet and avoid the case of having the claim stated within an image or video.

2. Try to choose tweet examples that exhibit hard matching with the text of the claim.

3. Exclude tweets in case it is not clear whether they are about the claim.

4. Exclude tweets containing multiple claims.

The collection of verified claims was collected from the AraFacts dataset [33] and a translated version of the ClaimsKG English dataset [34].

Since the AraFacts dataset is collected from five fact-checking organizations, there is a possibility that a claim might be mentioned multiple times in the dataset with different forms. Therefore, a given tweet might be linked to multiple claims. To tackle this case, we follow two steps. First, we group claims that have Jaccard similarity above 30% and exclude the ones that are non-similar. Second, we index the verified claim collection and retrieve the top-25 potential relevant claim for each tweet in gold pairs. Then, we expand the gold pairs with the missing claims accordingly.

After applying the previously mentioned guidelines, we ended up with 858 gold pairs.

CHAPTER 4: EXPERIMENTAL SETUP

In this section, we introduce the strategies followed by our experiments. We explain the used datasets, the adopted retrieval models and the baselines, and the selected BERT variants. Moreover, we elaborate on the applied evaluation measures, tools used for preprocessing, and on the method of building the training set.

4.1. Datasets

We conducted experiments on three claim retrieval datasets, namely CheckThat! 2020 English (CT2020-En), CheckThat! 2021 English (CT2021-En), and Arabic (CT2021-Ar) datasets. **CT2020-En** [8] is the official dataset for the CheckThat! 2020 lab. The verified claims are collected from Snopes, a well-known fact-checking website, and the queries are crawled tweets that were cited in articles debunking the rumors. **CT2021-En** [9] is an extension of CT2020-En, with a larger number of verified claims and queries. **CT2021-Ar** [9] is the official Arabic dataset for the CheckThat! 2021 lab that we contributed to its creation. Table 4.1 shows statistics about the three datasets. We notice that a query in CT2021-Ar can have more than one relevant verified claim.

4.2. Initial Retrieval and Preprocessing

To form the initial ranked set, we experimented with multiple classical retrieval approaches. BM25, Jelinek-Mercer smoothing, RM3, and DPH are the models we explored for this task. We constructed the index and retrieval models using PyTerrier library [35].

As preprocessing steps, we used Twitter API to get usernames out of user handles. Besides, we utilized Meta Reverse Image Search API(MRISA) [36] to perform a reverse

Table 4.1: Size of the datasets used in our experiments. The values in parentheses indicate the average number of relevant verified claims per query.

Dataset	Train	Validation	Test	Verified claims
CT2020-En	800 (1)	197 (1)	200 (1)	10,375
CT2021-En	999 (1)	200 (1)	202 (1)	13,835
CT2021-Ar	512 (1.2)	85 (1.2)	261 (1.3)	30,329

image search for pictures and videos.

4.3. Baselines

For each dataset in CheckThat! lab, we chose the winning team as a strong baseline to compare against. For the English models, they are **Buster.AI** [17] and **Aschern** [23] from CheckThat! lab 2020 and 2021, respectively. For Arabic, we consider our submission to ChecktThat! lab 2021 lab [9] as a strong baseline since it provided more than 10 points enhancement over the organizers' baseline. We refer to this submission with **bigIR**.

4.4. Evaluation Measures and Significance Test

We follow the evaluation procedure adopted by CheckThat! lab, which considers Mean Average Precision at depth 5 (MAP@5) as the main evaluation measure. Furthermore, for the English experiments, we report Precision@1 (P@1) and Mean Reciprocal Rank (MRR) since the average number of relevant documents for a query in these datasets is 1. However, for Arabic experiments, we add the R-Precision (RP) measure to

account for queries that have multiple relevant documents. Finally, we also report Recall at depth 100 (R@100) to show an upper bound performance for the initial retrieval stage.

To test for statistical significance for the reported results, we applied the two-tailed paired t-test significance test for all evaluation measures. We report p-values and highlight the ones that are beyond the 5% significance level. As the computation of the test requires per-query results, we contacted the authors of the adopted baselines to get their submitted runs to CheckThat! Lab. For Arabic, we already have the run of the top team; however, for English, we got only the runs of the top team in CT2021-En and the second team in CT2020-En. In other words, only the run for the winning team in CT2020-En is missing. To alleviate this problem, we consider the model that is closest in performance to the winning team as a proxy baseline for the purpose of conducting the significance test.

4.5. BERT variants and Fine-tuning

After the introduction of BERT [10], many transformer-based pre-trained language models were proposed in the literature to address specific tasks, albeit several were not tried for the claim retrieval task. For English experiments, we studied multiple variants, namely, BERT [10], MPNet [37][1], RoBERTa[18][2], Multinligual-MPNet[37], and MiniLM [38].[3] Apart from the vanilla BERT, the choice of other models is attributed to their reported performance within SBERT leaderboard on diverse tasks from different domains. [4]

For the Arabic experiments, we surveyed several top-performing models compared

[1]We namely use STSb-MPNet and Paraphrase-MPNet
[2]https://huggingface.co/sentence-transformers/msmarco-roberta-base-v2
[3]https://huggingface.co/sentence-transformers/paraphrase-MiniLM-L12-v2
[4]https://www.sbert.net/docs/pretrained_models.html

by a recent benchmark paper [39] about the Arabic-based transformer models, namely, Arabic_BERT [40], GigaBERT [41], MARBERT [42], AraBERT [43], QARiB [44], and Arabic-ALBERT [45]. For all BERT variants, we chose the base version due to our limited GPU capacity. Generally, we adopt the versions that have been pre-trained on datasets close to the task we work on.

Following, we provide a little more information about each model:

- **MPNet** was proposed by Song et al. [37] and trained using masked and permuted language modeling techniques. While masked language modeling is the technique used to train BERT, permuted language modeling is the adopted strategy to pre-train XLNet [46]. Thus, MPNet leverages the advantages of BERT and XLNet and avoids their limitations. MPNet surpasses multiple pre-trained models (such as BERT, and RoBERTa, etc.) on a wide range of down-streaming tasks like GLUE [47] (General Language Understanding Evaluation benchmark) and SQuAD [48] (Stanford Question Answering Dataset). We experiment with two versions of this model. First of which was trained on Semantic Textual Similarity benchmark (STSb) [49] and second of which was pre-trained on paraphrase dataset. [5] Thus, we used STSb-MPNet [6] and Paraphrase-MPNet. [7]

- **RoBERTa** [18] which is a Robustly Optimized BERT Pretraining Approach. RoBERTa has been used widely on a variety of NLP tasks (such as GLUE, RACE, etc.) and showed high performance. We utilized the version which was pre-trained on MSMACRO dataset [8], namely Msmacro-RoBERTa. [9]

[5] https://www.sbert.net/examples/training/paraphrases/README.html
[6] https://huggingface.co/sentence-transformers/stsb-mpnet-base-v2
[7] https://huggingface.co/sentence-transformers/paraphrase-mpnet-base-v2
[8] https://github.com/microsoft/MSMARCO-Passage-Ranking
[9] https://huggingface.co/sentence-transformers/msmarco-roberta-base-v2

- **Multilingual-MPNet** is a multilingual version of MPNet, which was pre-trained on Paraphrase dataset.

- **MiniLM** is a small and fast pre-trained model designed to tackle language understanding and generation tasks [38]. We adopt the version pre-trained on paraphrase dataset. [10]

- **GigaBERT** [41] is a bilingual model for English and Arabic. It was trained on a corpus collected from Gigaword, Oscar, and Wikipedia of a total of 10B tokens.

- **MARBERT** [42] is an Arabic BERT-based model. To improve the model's ability to process dialectal Arabic, the authors set 50% of the training data to be Arabic tweets. The total training size is 128 GB.

- **AraBERT** is the first Arabic-specific BERT model proposed by Antoun et al. [43]. We use the 0.2 version, which was trained on 200M sentences (77GB of text).

- **QARiB** introduced by Chowdury et al. [44] is new Arabic model. Toward improving the generalization of the model, the authors built a training set of both formal and informal data like news articles and tweets. They included 120M sentences and tweets in the training corpus.

- **Arabic_BERT** introduced by Safay et al. [40] . They trained it on a corpus comprising an unshuffled Arabic version of OSCAR data and Arabic Wikipedia with a total of 95GB of text.

- **Arabic-ALBERT** provided by [45] as an Arabic Version of ALBERT [50]. Similar to Arabic_BERT, it was trained on unshuffled Arabic version OSCAR data

[10] `https://huggingface.co/sentence-transformers/paraphrase-MiniLM-L12-v2`

and Arabic Wikipedia. Arabic ALBERT is the fastest model compared with other Arabic models [39].

Based on BERT authors' recommendations [10], for all BERT-based experiments, we tuned the following hyper-parameters on the dev set: number of epochs (either 2, 3, or 4), and learning rate (2e-5 or 3e-5). In addition, we employed one classification layer on top of BERT variants with two neurons as an output. For the dropout hyperparameter in the classification layer, after trying multiple values over a single BERT-based model with different random seeds, we find that the stable values are (0.3 or 0.4). Therefore, we adopt these values for tuning the other models. We fine-tuned each model five times with different random seeds, and we reported the median performance of those runs. The median is chosen to get a more logical value as some runs might be stuck in a local minimum or maximum.

4.6. Building the Training Set

To fine-tune BERT-based models, we constructed a balanced training set of positive and negative query-document pairs. The positive pairs are formed by pairing the query with its relevant verified claim and also the title of the verified claim (which constitutes a summary of it). For the negative pairs, we noticed that many methods in the literature adopt the sampling of "hard" pairs (i.e., the ones that are potential false positives) for tuning the reranker. However, that exposes the model only to one kind of pairs, with no exposure to other potentially-easy ones. As the reranker, at inference time, would be applied to all kinds of pairs, we choose the negative pairs randomly from the top-k irrelevant documents of the initial retrieved list. As a result, the chosen pairs cover a wider range of difficulties, allowing the model to be more robust.

4.7. Curriculum Learning Setup

The setup for the curriculum learning experiment is similar to that in the monoBERT experiment. The difference here is that we only tune two hyperparameters. The curriculum threshold m is the first, and the number of epochs is the second. While, for m (curriculum threshold), we tried 2, 3, 4, and 5, we chose 4, 5, and 6 to tune the number of epochs hyperparameter. The values of other hyperparameters are attained from monoBERT experiment. We chose the top-3 monoBERT performed models to perform this experiment.

4.8. DuoBERT Setup

We built the training set for duoBERT by exploiting the best monoBERT reranker. A positive training sample for a duoBERT reranker consists of a triplet; query, relevant verified claim, and non-relevant verified claim. The negative sample represents the opposite case, i.e., (query, non-relevant verified claim, relevant verified claim). After excluding the relevant verified claims, we chose the non-relevant verified claim randomly from top-k verified claims reranked by monoBERT. Therefore, k is a hyperparameter that we tuned on the dev set; specifically, we used 10 and 20. Another hyperparameter is the number of triplets for the same query. We tuned this hyperparameter on 4, 6, and 8 values. The relevant verified claim could be its description or title. We adopted the same strategy applied in the curriclum learning experiment by tuning only the top-3 performed models in the monoBERT experiment. For development and testing sets, we choose five as the size of the input list for the duoBERT reranker to alleviate the high cost of applying duoBERT.

CHAPTER 5: EXPERIMENTAL EVALUATION AND RESULTS

In our experiments, we aim to answer the following research questions:

RQ1 What is the impact of the preprocessing steps on the performance of the initial retrieval stage? (Section 5.1)

RQ2 Does a monoBERT reranker improve the performance over the initial retrieval stage? What is the best BERT variant for the task? (Sections 5.2)

RQ3 How can we effectively leverage the title and the description of verified claims in training and inference? (Section 5.3)

RQ4 Does applying curriculum learning in claim retrieval setup enhance the performance? (Section 5.4)

RQ5 Does duoBERT technique provide an improvement over the point-wise approaches? (Section 5.5)

RQ6 What will the performance of the proposed approach be on Arabic data? (Section 5.6)

5.1. Initial Retrieval with Preprocessing Experiment (**RQ1**)

To answer **RQ1**, we apply preprocessing with multiple classical retrieval models, namely BM25[31], Jelink-Mercer (JM) smoothing [51], DPH [52], and RM3 [53].

We conducted the experiment on both CT2020-En and CT2021-En. Table 5.1 shows the performance of each of those models before and after applying preprocessing over the dev-set of CT2020-En. We omit the results on CT2021-En as they exhibit very similar performance. The results show that BM25 is superior to the other models for our task.

Table 5.1: Performance of the initial retrieval stage on CT2020-En-dev before and after applying preprocessing (PreP).

Model	MAP@5 (p-value)	P@1 (p-value)	MRR (p-value)	R@100 (p-value)
BM25	0.710 (baseline)	0.594 (baseline)	0.717 (baseline)	0.949 (baseline)
BM25+PreP	**0.733 (0.076)**	**0.609 (0.319)**	**0.739 (0.067)**	**0.954 (0.565)**
DPH	0.685 (baseline)	0.563 (baseline)	0.692 (baseline)	0.939 (baseline)
DPH+PreP	0.721 (0.008)*	0.599 (0.034)*	0.727 (0.007) *	**0.954 (0.083)**
JM	0.687 ((baseline)	0.558 (baseline)	0.695 (baseline)	0.944 (baseline)
JM+PreP	0.712 (0.066)	0.579 (0.207)	0.719 (0.062)	0.944 (1.000)
RM3	0.692 (baseline)	0.594 (baseline)	0.697 (baseline)	0.898 (baseline)
RM3+PreP	0.713 (0.141)	**0.609 (0.319)**	0.719 (0.122)	0.914 (0.180)

More importantly, the table clearly shows the effectiveness of applying the preprocessing step, as the performance improves for all models over all measures. Therefore, we adopt BM25 with preprocessing for the initial retrieval for the remaining experiments. We also notice that the best performance reaches P@1 of about 61%, which leaves a large room for potential improvement for the reranking stage. Furthermore, reaching 95% recall at depth 100 indicates that the improvement is indeed possible by an effective reranker.

5.2. Reranking using monoBERT (**RQ2**)

As rerankers are more complex, thus expensive, it is critical to first choose a *short* but *effective* depth of the initial retrieved list of claims to be reranked. We conducted an experiment to tune the depth using RoBERTa as an example reranker that demonstrated high performance for our task in the literature [17]. Table 5.2 illustrates the performance

Table 5.2: Performance of MSMarco-RoBERTa on CT2020-En-dev with varying depth of the initial retrieval set retrieved by BM25.

Depth	MAP@5	P@1	MRR
10	0.874	0.848	0.875
20	**0.881**	**0.853**	**0.881**
30	0.876	0.843	0.877
50	0.874	0.838	0.875
100	0.867	0.822	0.871

of RoBERTa as a monoBERT reranker for different depth values of the initial retrieved list (10, 20, 30, 50, and 100). The experiment shows that a depth of 20 exhibits the best performance among the different experimented values; therefore, we stick to it in the rest of the experiments. Moreover, we observe that reranking using a BERT-based model gives a considerable improvement over the performance of the initial retrieval models (85% vs. 61% in P@1), as illustrated earlier in Table 5.1, which highlights the importance of using contextualized models for this task.

We next turn to answer **RQ2** by comparing the performance of multiple BERT-based models with the adopted baselines on the two English datasets. Tables 5.4 and 5.3 present the performance of the models over the test sets of CT2020-En, and CT2021-En respectively.

The results show that two models, namely STSb-MPNet and Paraphrase-MPNet, outperform the two baselines for CT2020-En, but only STSb-MPNet is significantly better. As for CT2021-En, four models, namely STSb-MPNet, Paraphrase-MPNet,

Table 5.3: Performance of monoBERT models on the test set of CT2021-En with corresponding p-value for each metric with parenthesis. We add aster-stick next to p-values below 5%

Model	MAP@5 (p-val)	P@1 (p-val)	MRR (p-val)
Aschern (Best at CheckThat! lab 2021)	0.883	0.861	0.884
Multilingual-MPNet	0.742 (0.000)*	0.644 (0.000)*	0.749 (0.000)*
BERT	0.834 (0.055)	0.757 (0.002)*	0.835 (0.054)
MiniLM	0.904 (0.371)	0.871 (0.725)	0.906 (0.344)
MSMarco-RoBERTa	0.916 (0.122)	0.876 (0.579)	0.917 (0.133)
STSb-MPNet (Best model of 2020)	0.917 (0.124)	0.876 (0.603)	0.918 (0.132)
Paraphrase-MPNet	0.922 (0.067)	0.886 (0.337)	0.923 (0.071)
STSb-MPNet	**0.929 (0.030)** *	**0.901 (0.103)**	**0.929 (0.035)***

MSMacro-RoBERTa, and MiniLM, outperform the best-performing team in the CheckThat! lab 2021, as indicated over all measures. However, STSb-MPNet is the only significant model. We test the best model attained for CT2020-En (STSb-MPNet 2020) on CT2021-En without any changes and notice that it also surpasses the state-of-the-art model. The vanilla BERT model exhibits poor performance on both datasets compared to other models. Additionally, we notice that the used multi-lingual model seems to be unsuitable for this kind of task.

5.3. Leveraging Verified Claim Fields (**RQ3**)

In all of the previous experiments, we used both VClaim and Title separately as training examples and performed inference using VClaim only, as done in [19]. However, in this experiment, we probe the impact of using other combinations of those fields on

Table 5.4: Performance of monoBERT models on the test set of CT2020-En. While the first value between parenthesis refers to *p*-value for the first baseline (proxy for the top team which is MSMarco-RoBERTa), the second value refers to *p*-value for the second baseline (second team which is UNIPI). We add aster-stick for *p*-values below 5% with respect to baseline1 and plus sign for the baseline2 case.

Model	MAP@5 (*p*-val1, *p*-val2)	P@1 (*p*-val1, *p*-val2)	MRR (*p*-val1, *p*-val3)
Buster.AI (Best at CheckThat! 2020)	0.929	0.895	0.927
MSMarco-RoBERTa (baseline1)	0.926	0.894	0.926
UNIPI (baseline2)	0.914	0.879	0.915
Multilingual-MPNet	0.650 (0.000, 0.000)$^{*+}$	0.553 (0.000, 0.000)$^{*+}$	0.666 (0.000, 0.000)$^{*+}$
BERT	0.735 (0.000, 0.000)$^{*+}$	0.618 (0.000, 0.000) $^{*+}$	0.741 (0.000, 0.000)$^{*+}$
MiniLM	0.920 (0.593, 0.707)	0.884 (0.656, 0.848)	0.920 (0.593, 0.780)
Paraphrase-MPNet	0.944 (0.128, 0.076)	0.925 (0.158, 0.072)	0.944 (0.128, 0.091)
STSb-MPNet	**0.955 (0.003, 0.011)**$^{*+}$	**0.950 (0.002, 0.003)**$^{*+}$	**0.955 (0.003, 0.014)** $^{*+}$

the performance. To answer **RQ3**, we experiment with training using VClaim only, Title only, and both VClaim and Title. We also experiment using both VClaim and Title at inference, where the relevance score of a claim is the average score of tweet-VClaim and tweet-Title pairs.

We conducted such an experiment on CT2021-En. We chose the top three performed in the previous experiment, namely, STSb-MPNet, Paraphrase-MPNet, and MSMarco-RoBERTa. Table 5.5 presents the performance using the different combinations. We notice that using both VClaim and Title for training and inference yields the best performance in all models, with a statistically significant difference with respect to the title-only baselines.

We believe training on both increases the model's understanding of the claim context. Moreover, the training set size is doubled when we add Title in training.

Table 5.5: Performance using Verified Claim (VCl) and Title (Ttl) fields of the verified claim on CT2021-En. We report *p*-value for MAP@5 measure only and add aster-stick next to *p*-values below 5%

Model	Training	Inference	MAP@5 (*p*-value)	P@1 (*p*-value)	MRR (*p*-value)
Paraphrase-MPNet	Ttl	Ttl	0.840 (baseline)	0.762 (baseline)	0.841 (baseline)
	VCl	VCl	0.893 (0.003) *	0.842 (0.006) *	0.893 (0.003) *
	VCl+Ttl	VCl	0.922 (0.000) *	0.886 (0.000) *	0.923 (0.000) *
	VCl+Ttl	VCl+Ttl	0.926 (0.000) *	0.891 (0.000) *	0.927 (0.000) *
MSMarco-RoBERTa	Ttl	Ttl	0.840 (baseline)	0.762 (baseline)	0.841 (baseline)
	VCl	VCl	0.882 (0.024)*	0.837 (0.022)*	0.883 (0.025) *
	VCl+Ttl	VCl	0.916 (0.000)*	0.876 (0.000)*	0.917 (0.000)*
	VCl+Ttl	VCl+Ttl	0.920 (0.000)*	0.881 (0.000)*	0.920 (0.000)*
STSb-MPNet	Ttl	Ttl	0.884 (baseline)	0.842 (baseline)	0.886 (baseline)
	VCl	VCl	0.908 (0.095)	0.876 (0.145)	0.909 (0.105)
	VCl+Ttl	VCl	0.929 (0.006) *	0.901 (0.014) *	0.929 (0.007) *
	VCl+Ttl	VCl+Ttl	**0.936 (0.002)***	**0.911 (0.003)***	**0.936 (0.002)***

Table 5.5 reveals the evaluation results of using different combinations of VerClaim and Title during the training and inference. In one of the combinations, we opt to train using both VerClaim and Title but test using VerClaim only to check whether adding the Titles during training improves performance during inference. We notice that using both VerClaim and Title for training and testing yields the best performance. Looking at rows that have VerClaim and Title in training, we observe nearly one-point improvement for all models by just adding the score of tweet-Title pair during the inference phase. While using just VerClaim or Title only leads to performance degradation, using both provides remarkable enhancements. The reason could be attributed to two factors; first,

adding Title along with VerClaim increases the model's chance to understand the claim as their combination exhibits the claim in two different forms. Second, the training set size is doubled when we add Title in the training phase.

5.4. Employing Curriculum Learning in MonoBERT Reranker (**RQ4**)

Table 5.6 and 5.7 provide the evaluation results for top-3 performed models on CT2021-En and CT2020-En respectively. We notice that employing curriculum learning technique during training monoBERT did not provide an enhancement for all models. The reason could be attributed to the fact the used collection is considered small; therefore, the variance between hard and easy examples is small as well. We believe that applying the proposed function on a bigger and yet more challenging dataset would provide potential improvements.

5.5. Reranking using DuoBERT (**RQ5**)

Looking at Table 5.6 and 5.7, we observe that duoBERT provided a little improvement over monoBERT baseline for two models in CT-2021-En. However, duoBERT was worse than monoBERT over all models for CT-2020-En. The reason for having such a difference in performance is that CT-2020-En training set size is smaller than that in CT-2021-En. Trying this technique over a more challenging dataset is a promising potential and one of our future plans.

5.6. Performance on Arabic Data (**RQ6**)

To answer **RQ6**, we examine the effectiveness of the proposed pipeline by applying the attained conclusions from English experiments on the Arabic dataset. More

Table 5.6: Performance of applying curriculum learning technique and duoBER on the test set of CT2020-En. We report the significance test p-value for each measure between parenthesis and add aster-stick next to those below 5%

Model	MAP@5 (p-value)	P@1 (p-value)	MRR (p-value)
mono-STSb-MPNet	**0.955 (baseline)**	**0.950 (baseline)**	**0.955 (baseline)**
mono-STSb-MPNet+curriculum	0.922 (0.002) *	0.899 (0.004)*	0.924 (0.002)*
duo-STSb-MPNet	0.940 (0.047) *	0.930 (0.103)	0.940 (0.047) *
mono-MSMarco-RoBERTa	0.926 (baseline)	0.894 (baseline)	0.926 (baseline)
mono-MSMarco-RoBERTa+curriculum	0.927 (0.953)	0.899 (0.797)	0.928 (0.890)
duo-MSMarco-RoBERTa	0.896 (0.054)	0.849 (0.095)	0.896 (0.054)
mono-Paraphrase-MPNet	0.944 (baseline)	0.925 (baseline)	0.944 (baseline)
mono-Paraphrase-MPNet+curriculum	0.905 (0.005) *	0.869 (0.016)*	0.907 (0.006)*
duo-Paraphrase-MPNet	0.931 (0.255)	0.915 (0.594)	0.931 (0.255)

specifically, we performed three steps.

(1) We conducted the same preprocessing steps applied to English. We then experimented with multiple classical models for the initial retrieval phase. As shown in Table 5.8, BM25 is found to be the best-performing model for all evaluation measures except for R@100. However, looking at other depths of recall, BM25 was the best. Therefore, BM25 was adopted to retrieve the initial ranked list in subsequent experiments.

(2) We chose AraBERT (as it is the model used by the adopted baseline in CheckThat 2020 [9]) to tune the depth of the initial retrieval set. Similar to English experiments, we ran AraBERT model on different depth values of the initial retrieved list (10, 20, 30, 50, and 100). From Table 5.9, we notice that the best performance is observed when the

Table 5.7: Performance of applying curriculum learning technique and duoBERT on the test set of CT2021-En. We report the significance test p-value for each measure between parenthesis and add aster-stick next to those below 5%

Model	MAP@5 (p-value)	P@1 (p-value)	MRR (p-value)
mono-STSb-MPNet	**0.936 (baseline)**	**0.911 (baseline)**	**0.936 (baseline)**
mono-STSb-MPNet+curriculum	0.924 (0.320)	0.891 (0.319)	0.925 (0.328)
duo-STSb-MPNet	0.931 (0.603)	0.896 (0.367)	0.931 (0.603)
mono-MSMarco-RoBERTa	0.920 (baseline)	0.881 (baseline)	0.920 (baseline)
mono-MSMarco-RoBERTa+curriculum	0.911 (0.317)	0.871 (0.528)	0.912 (0.334)
duo-MSMarco-RoBERTa	0.926 (0.556)	0.881 (1.000)	0.926 (0.556)
mono-Paraphrase-MPNet	0.926 (baseline)	0.891 (baseline)	0.927 (baseline)
mono-Paraphrase-MPNet+curriculum	0.910 (0.052)	0.861 (0.058)	0.910 (0.052)
duo-Paraphrase-MPNet	0.930 (0.611)	0.896 (0.740)	0.930 (0.667)

depth is set to 30 with a value reaching 93% for MAP@5.

(3) We experimented with the top-performing Arabic BERT-based models as monoBERT rerankers. For all models, we exploited both VClaim and Title fields during training and inference.

Table 5.10 shows the performance of different BERT-based Arabic models on the test set of CT2021-Ar. Both AraBERT and GigaBERT outperform the baseline for all measures, with statistically significant improvement in MAP@5 and MRR. Moreover, two additional models, Arabic-ALBERT and Arabic-BERT, exhibit better performance over the baseline.

Table 5.8: Performance of the initial retrieval stage on CT202-Ar-dev before and after applying preprocessing (PreP) with significance test p-value between parenthesis.

Model	MAP@5 (p-value)	P@1 (p-value)	RP (p-value)	MRR (p-value)	R@100 (p-value)
JM	0.812 (baseline)	0.800 (baseline)	0.776 (baseline)	0.839 (baseline)	0.943 (baseline)
JM+PreP	0.827 (0.144)	0.800 (1.000)	0.782 (0.567)	0.843 (0.367)	0.949 (0.320)
BM25	0.827 (baseline)	0.800 (baseline)	0.788 (baseline)	0.845 (baseline)	0.955 (baseline)
BM25+PreP	**0.848 (0.101)**	**0.824 (0.159)**	**0.812 (0.159)**	**0.860 (0.166)**	0.961 (0.320)
DPH	0.798 (baseline)	0.776 (baseline)	0.759 (baseline)	0.820 (baseline)	0.943 (baseline)
DPH+PreP	0.820 (0.098)	0.800 (0.159)	0.782 (0.103)	0.841 (0.164)	0.949 (0.320)
RM3	0.716 (baseline)	0.600 (baseline)	0.598 (baseline)	0.735 (baseline)	0.976 (baseline)
RM3+PreP	0.712 (0.712)	0.600 (1.000)	0.592 (0.657)	0.728 (0.444)	**0.976 (1.000)**

Table 5.9: Performance of AraBERT on CT2021-Ar-dev with varying depth of the initial retrieval stage.

Depth	MAP@5	P@1	MRR	RP
10	0.894	0.894	0.900	0.886
20	0.900	0.906	0.912	0.894
30	**0.935**	**0.941**	**0.947**	**0.929**
50	0.925	0.918	0.935	0.912
100	0.925	0.918	0.935	0.912

Table 5.10: Evaluation of BERT-based Arabic models on CT2021-Ar. We report the significance test *p*-value for each measure between parenthesis and add aster-stick next to those below 5%

Model	MAP@5 (*p*-value)	P@1 (*p*-value)	MRR (*p*-value)	RP (*p*-value)
bigIR (Best at the CheckThat! lab 2021)	0.908	0.908	0.924	0.895
MARBERT	0.767 (0.000)*	0.743 (0.000)*	0.813 (0.000) *	0.707 (0.000) *
QARiB	0.903 (0.711)	0.885 (0.240)	0.924 (0.955)	0.861 (0.085)
Arabic-ALBERT	0.921 (0.347)	0.923 (0.395)	0.948 (0.083)	0.898 (0.870)
Arabic_BERT	0.932 (0.074)	0.935 (0.108)	0.956 (0.015) *	0.910 (0.375)
GigaBERT-v3	0.939 (0.021)*	0.939 (0.059)	0.956 (0.020)*	0.918 (0.152)
AraBERT-	**0.940 (0.023)** *	**0.946 (0.033)** *	**0.959 (0.012)***	**0.927 (0.060)**

CHAPTER 6: CONCLUSIONS AND FUTURE WORK

6.1. Conclusions

In this thesis, we proposed a pipeline to retrieve previously fact-checked claims with high effectiveness. We convert the ambiguous content in the queries such as URLs, images, and Twitter handles in queries to helpful data. In addition, we employ a powerful BERT-variant as a point-wise reranker. Then, we study the impact of employing different fields of the verified claim during the training and the testing processes. Additionally, we proposed a novel difficulty function within the setup of curriculum learning and exploited duoBERT technique for this task as well. We create the first Arabic dataset for the claim retrieval task. Furthermore, we are the first to develop a system for this task in Arabic. The experiments show that the proposed pipeline outperforms the state-of-the-art by a noticeable margin and yet with a simpler approach. Not only does the proposed method outperform the state of the art in English, but it also shows high effectiveness when it was applied on the Arabic dataset, indicating that it is a promising setup for this task on multiple languages.

6.2. Publications

The following are notable contributions made by this thesis work:

1. W. Mansour, T. Elsayed, and A. Al-Ali, "Did I See it Before? Detecting Previously-Checked Claims over Twitter" in Proceedings of the 44th European Conference on Information Retrieval, 2022 [54].

2. Z. S. Ali, W. Mansour, T. Elsayed, and A. Al-Ali, "Arafacts: The first large Arabic dataset of naturally occurring claims," in Proceedings of the Sixth Arabic Natural

Language Processing Workshop, 2021, pp. 231–236 [33].

3. S. Shaar, F. Haouari, W. Mansour, M. Hasanain, N. Babulkov, F. Alam, G. Da SanMartino, T. Elsayed, and P. Nakov, "Overview of the CLEF-2021 CheckThat! lab task2 on detecting previously fact-checked claims in tweets and political debates," in CLEF 2021 Working Notes. Working Notes of CLEF 2021–Conference and Labs of the Evaluation Forum, G. Faggioli, N. Ferro, A. Joly, M. Maistro, and F. Piroi,Eds., ser. CEUR Workshop Proceedings, CEUR-WS.org, 2021 [9].

4. P. Nakov, D. S. M. Giovanni, T. Elsayed, A. Barrón-Cedeño, R. Míguez, S.Shaar, F. Alam, F. Haouari, M. Hasanain, W. Mansour, B. Hamdan, Z. S. Ali,N. Babulkov, A. Nikolov, G. K. Shahi, J. M. Struß, T. Mandl, M. Kutlu, and Y. S. Kartal, "Overview of the CLEF-2021 CheckThat! lab on detecting check-worthy claims, previously fact-checked claims, and fake news," in Proceedings of the Twelfth International Conference of the CLEF Association: Experimental IR Meets Multilinguality, Multimodality, and Interaction, ser. CLEF 2021, 2021 [15].

6.3. Future Work

Despite having a lot of progress regarding the claim retrieval problem, there is still ample room for improvement in the following. One limitation of this work is assuming that there is a previously fact-checked claim for a given input. However, in some cases, the input claim does not have any relevant claim in the collection of previously verified claims. Therefore, we plan to build a system capable of predicting whether there is a previously verified claim for a given input claim.

Also, the current system works on a static dataset. However, there are many new

claims added every day over fact-checking websites. To address this limitation, we plan to build a periodically updated dataset that collects fact-checked claims from multiple authorized resources. Then, we can deploy the proposed system into a real-time application which in turn utilizes the updated dataset and provides up-to-date predictions.

Finally, since our system supports Arabic and English languages only, we target to build a system that addresses other languages too.

REFERENCES

[1] R. Gunther, P. A. Beck, and E. C. Nisbet, "Fake news did have a significant impact on the vote in the 2016 election: Original full-length version with methodological appendix," *Unpublished manuscript, Ohio State University, Columbus, OH*, 2018.

[2] M. Broersma and T. Graham, "Social media as beat: Tweets as a news source during the 2010 british and dutch elections," *journalism practice*, vol. 6, no. 3, pp. 403–419, 2012.

[3] K. Shu, A. Sliva, S. Wang, J. Tang, and H. Liu, "Fake news detection on social media: A data mining perspective," *ACM SIGKDD explorations newsletter*, vol. 19, no. 1, pp. 22–36, 2017.

[4] J. Thorne and A. Vlachos, "Automated fact checking: Task formulations, methods and future directions," in *Proceedings of the 27th International Conference on Computational Linguistics*, 2018, pp. 3346–3359.

[5] S. Vosoughi, D. Roy, and S. Aral, "The spread of true and false news online," *Science*, vol. 359, no. 6380, pp. 1146–1151, 2018.

[6] T. Elsayed, P. Nakov, A. Barrón-Cedeno, M. Hasanain, R. Suwaileh, G. Da San Martino, and P. Atanasova, "Checkthat! at clef 2019: Automatic identification and verification of claims," in *European Conference on Information Retrieval*, Springer, 2019, pp. 309–315.

[7] N. Hassan, F. Arslan, C. Li, and M. Tremayne, "Toward automated fact-checking: Detecting check-worthy factual claims by ClaimBuster," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2017, pp. 1803–1812.

[8] A. Barrón-Cedeño, T. Elsayed, P. Nakov, G. Da San Martino, M. Hasanain, R. Suwaileh, F. Haouari, N. Babulkov, B. Hamdan, A. Nikolov, S. Shaar, and Z. Ali, "Overview of CheckThat! 2020 — automatic identification and verification of claims in social media," in *Proceedings of the 11th International Conference of the CLEF Association: Experimental IR Meets Multilinguality, Multimodality, and Interaction*, ser. CLEF '2020, 2020, pp. 215–236.

[9] S. Shaar, F. Haouari, W. Mansour, M. Hasanain, N. Babulkov, F. Alam, G. Da San Martino, T. Elsayed, and P. Nakov, "Overview of the CLEF-2021 CheckThat! lab task 2 on detecting previously fact-checked claims in tweets and political debates," in *CLEF 2021 Working Notes. Working Notes of CLEF 2021–Conference and Labs of the Evaluation Forum*, G. Faggioli, N. Ferro, A. Joly, M. Maistro, and F. Piroi, Eds., ser. CEUR Workshop Proceedings, CEUR-WS.org, 2021.

[10] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 2019, pp. 4171–4186.

[11] Google, *Fact check*. [Online]. Available: `https://toolbox.google.com/factcheck/explorer`.

[12] S. Shaar, N. Babulkov, G. Da San Martino, and P. Nakov, "That is a known lie: Detecting previously fact-checked claims," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 3607–3618.

[13] N. Vo and K. Lee, "Where Are the Facts? Searching for fact-checked information to alleviate the spread of fake news," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020, pp. 7717–7731.

[14] Q. Sheng, J. Cao, X. Zhang, X. Li, and L. Zhong, "Article reranking by memory-enhanced key sentence matching for detecting previously fact-checked claims," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, 2021, pp. 5468–5481.

[15] P. Nakov, D. S. M. Giovanni, T. Elsayed, A. Barrón-Cedeño, R. Míguez, S. Shaar, F. Alam, F. Haouari, M. Hasanain, W. Mansour, B. Hamdan, Z. S. Ali, N. Babulkov, A. Nikolov, G. K. Shahi, J. M. Struß, T. Mandl, M. Kutlu, and Y. S. Kartal, "Overview of the CLEF-2021 CheckThat! lab on detecting check-worthy claims, previously fact-checked claims, and fake news," in *Proceedings of the Twelfth International Conference of the CLEF Association: Experimental IR Meets Multilinguality, Multimodality, and Interaction*, ser. CLEF 2021, 2021.

[16] S. Shaar, F. Alam, G. D. S. Martino, and P. Nakov, "The role of context in detecting previously fact-checked claims," *arXiv preprint arXiv:2104.07423*, 2021.

[17] M. Bouziane, H. Perrin, A. Cluzeau, J. Mardas, and A. Sadeq, "Buster.AI at CheckThat! 2020: Insights and recommendations to improve fact-checking.," in *CLEF 2020 Working Notes*, L. Cappellato, C. Eickhoff, N. Ferro, and A. Névéol, Eds., ser. CEUR Workshop Proceedings, CEUR-WS.org, 2020.

[18] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized bert pretraining approach," *arXiv preprint arXiv:1907.11692*, 2019.

[19] L. Passaro, A. Bondielli, A. Lenci, and F. Marcelloni, "UNIPI-NLE at CheckThat! 2020: Approaching fact checking from a sentence similarity perspective through the lens of transformers," in *CLEF 2020 Working Notes*, L. Cappellato, C. Eickhoff, N. Ferro, and A. Névéol, Eds., ser. CEUR Workshop Proceedings, CEUR-WS.org, 2020.

[20] N. Reimers and I. Gurevych, "Sentence-Bert: Sentence embeddings using Siamese Bert-networks," *arXiv preprint arXiv:1908.10084*, 2019.

[21] E. Thuma, M. N. Peace, L.-D. Tebo, and M. Monkgogi, "UB ET at CheckThat! 2020: Exploring ad hoc retrieval approaches in verified claims retrieval," in *CLEF 2020 Working Notes*, L. Cappellato, C. Eickhoff, N. Ferro, and A. Névéol, Eds., ser. CEUR Workshop Proceedings, CEUR-WS.org, 2020.

[22] T. McDonald, Z. Dong, Y. Zhang, R. Hampson, J. Young, Q. Cao, J. Leidner, and M. Stevenson, "The University of Sheffield at CheckThat! 2020: Claim identification and verification on twitter," in *CLEF 2020 Working Notes*, L. Cappellato, C. Eickhoff, N. Ferro, and A. Névéol, Eds., ser. CEUR Workshop Proceedings, CEUR-WS.org, 2020.

[23] A. Chernyavskiy, D. Ilvovsky, and P. Nakov, "Aschern at CheckThat! 2021: Lambda-calculus of fact-checked claims," in *CLEF 2021 Working Notes. Working Notes of CLEF 2021–Conference and Labs of the Evaluation Forum*, G.

Faggioli, N. Ferro, A. Joly, M. Maistro, and F. Piroi, Eds., ser. CEUR Workshop Proceedings, CEUR-WS.org, 2021.

[24] A. Kazemi, K. Garimella, D. Gaffney, and S. Hale, "Claim matching beyond English to scale global fact-checking," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, 2021, pp. 4504–4517.

[25] T. F. Coleman and Z. Wu, "Parallel continuation-based global optimization for molecular conformation and protein folding," *Journal of Global Optimization*, vol. 8, no. 1, pp. 49–65, 1996.

[26] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum learning," in *Proceedings of the 26th annual international conference on machine learning*, 2009, pp. 41–48.

[27] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, "Natural language processing (almost) from scratch," *Journal of machine learning research*, vol. 12, no. ARTICLE, pp. 2493–2537, 2011.

[28] B. Hu, Z. Lu, H. Li, and Q. Chen, "Convolutional neural network architectures for matching natural language sentences," in *Advances in neural information processing systems*, 2014, pp. 2042–2050.

[29] A. Diba, V. Sharma, A. Pazandeh, H. Pirsiavash, and L. Van Gool, "Weakly supervised cascaded convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 914–922.

[30] S. MacAvaney, F. M. Nardini, R. Perego, N. Tonellotto, N. Goharian, and O. Frieder, "Training curricula for open domain answer re-ranking," in *Proceedings*

of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, 2020, pp. 529–538.

[31] S. Robertson and H. Zaragoza, "The probabilistic relevance framework: BM25 and Beyond," *Foundations and Trends in Information Retrieval*, vol. 3, no. 4, pp. 333–389, 2009.

[32] R. Nogueira, W. Yang, K. Cho, and J. Lin, "Multi-stage document ranking with bert," *ArXiv*, vol. abs/1910.14424, 2019.

[33] Z. S. Ali, W. Mansour, T. Elsayed, and A. Al-Ali, "AraFacts: The first large arabic dataset of naturally occurring claims," in *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, 2021, pp. 231–236.

[34] A. Tchechmedjiev, P. Fafalios, K. Boland, M. Gasquet, M. Zloch, B. Zapilko, S. Dietze, and K. Todorov, "ClaimsKG: A knowledge graph of fact-checked claims," in *International Semantic Web Conference*, Springer, 2019, pp. 309–324.

[35] C. Macdonald and N. Tonellotto, "Declarative experimentation in information retrieval using PyTerrier," in *Proceedings of ICTIR 2020*, 2020.

[36] R. Snowdon, *Meta reverse image search*, `https://github.com/vivithemage/mrisa`, 2017.

[37] K. Song, X. Tan, T. Qin, J. Lu, and T.-Y. Liu, "MPNet: Masked and permuted pre-training for language understanding," *arXiv preprint arXiv:2004.09297*, 2020.

[38] W. Wang, F. Wei, L. Dong, H. Bao, N. Yang, and M. Zhou, "MiniLM: Deep self-attention distillation for task-agnostic compression of pre-trained transformers," *arXiv preprint arXiv:2002.10957*, 2020.

[39] I. A. Farha and W. Magdy, "Benchmarking transformer-based language models for Arabic Sentiment and Sarcasm Detection," in *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, 2021, pp. 21–31.

[40] A. Safaya, M. Abdullatif, and D. Yuret, "Kuisail at SemEval-2020 task 12: Bert-CNN for offensive speech identification in social media," in *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, 2020, pp. 2054–2059.

[41] W. Lan, Y. Chen, W. Xu, and A. Ritter, "GigaBERT: Zero-shot Transfer Learning from English to Arabic," in *Proceedings of The 2020 Conference on Empirical Methods on Natural Language Processing (EMNLP)*, 2020.

[42] M. Abdul-Mageed, A. Elmadany, and E. M. B. Nagoudi, "ARBERT & MARBERT: deep bidirectional transformers for Arabic," *arXiv preprint arXiv:2101.01785*, 2020.

[43] W. Antoun, F. Baly, and H. Hajj, "AraBERT: Transformer-based Model for Arabic Language Understanding," in *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection*, 2020, pp. 9–15.

[44] S. A. Chowdhury, A. Abdelali, K. Darwish, J. Soon-Gyo, J. Salminen, and B. J. Jansen, "Improving Arabic text categorization using transformer training diversification," in *Proceedings of the Fifth Arabic Natural Language Processing Workshop*, 2020, pp. 226–236.

[45] A. Safaya, *Arabic-ALBERT*, version 1.0.0, Aug. 2020. DOI: 10.5281/zenodo.4718724.

[46] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. R. Salakhutdinov, and Q. V. Le, "Xlnet: Generalized autoregressive pretraining for language understanding," *Advances in neural information processing systems*, vol. 32, 2019.

[47] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. R. Bowman, "GLUE: A multi-task benchmark and analysis platform for natural language understanding," In the Proceedings of ICLR., 2019.

[48] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, "Squad: 100,000+ questions for machine comprehension of text," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 2016, pp. 2383–2392.

[49] D. Cer, M. Diab, E. Agirre, I. Lopez-Gazpio, and L. Specia, "Semeval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation," in *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, 2017, pp. 1–14.

[50] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut, "Albert: A lite bert for self-supervised learning of language representations," in *International Conference on Learning Representations*, 2019.

[51] F. Jelinek, "Interpolated estimation of Markov source parameters from sparse data," in *Proc. Workshop on Pattern Recognition in Practice, 1980*, 1980.

[52] G. Amati, G. Amodeo, M. Bianchi, C. Gaibisso, and G. Gambosi, "FUB, IASI-CNR and University of "Tor Vergata" at TREC 2008 Blog Track," in *The Seventeenth Text REtrieval Conference, TREC 2008*, US, 2008.

[53] V. Lavrenko and W. B. Croft, "Relevance based language models," in *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, 2001, pp. 120–127.

[54] W. Mansour, T. Elsayed, and A. Al-Ali, "Did I See it Before? Detecting Previously-Checked Claims over Twitter," in *Proceedings of the 44th European Conference on Information Retrieval*, 2022.

[55] G. Faggioli, N. Ferro, A. Joly, M. Maistro, and F. Piroi, Eds., *CLEF 2021 Working Notes. Working Notes of CLEF 2021–Conference and Labs of the Evaluation Forum*, CEUR Workshop Proceedings, CEUR-WS.org, 2021.

[56] L. Cappellato, C. Eickhoff, N. Ferro, and A. Névéol, Eds., *CLEF 2020 Working Notes*, CEUR Workshop Proceedings, CEUR-WS.org, 2020.