MythQA: Query-Based Large-Scale Check-Worthy Claim Detection through Multi-Answer Open-Domain Question Answering

Yang Bai
baiyang94@ufl.edu
The University of Florida
Gainesville, Florida, USA

Anthony Colas
acolas1@ufl.edu
The University of Florida
Gainesville, Florida, USA

Daisy Zhe Wang
daisyw@ufl.edu
The University of Florida
Gainesville, Florida, USA

ABSTRACT

Check-worthy claim detection aims at providing plausible misinformation to downstream fact-checking systems or human experts to check. This is a crucial step toward accelerating the fact-checking process. Many efforts have been put into how to identify check-worthy claims from a small scale of pre-collected claims, but how to efficiently detect check-worthy claims directly from a large-scale information source, such as Twitter, remains underexplored. To fill this gap, we introduce MythQA, a new multi-answer open-domain question answering(QA) task that involves contradictory stance mining for query-based large-scale check-worthy claim detection. The idea behind this is that contradictory claims are a strong indicator of misinformation that merits scrutiny by the appropriate authorities. To study this task, we construct TweetMythQA, an evaluation dataset containing 522 factoid multi-answer questions based on controversial topics. Each question is annotated with multiple answers. Moreover, we collect relevant tweets for each distinct answer, then classify them into three categories: "Supporting", "Refuting", and "Neutral". In total, we annotated 5.3K tweets. Contradictory evidence is collected for all answers in the dataset. Finally, we present a baseline system for MythQA and evaluate existing NLP models for each system component using the TweetMythQA dataset. We provide initial benchmarks and identify key challenges for future models to improve upon. Code and data are available at: https://github.com/TonyBY/Myth-QA

CCS CONCEPTS

• **Information systems** → **Question answering**; **Document filtering**; **Clustering and classification**; **Test collections**; **Relevance assessment**; **Retrieval effectiveness**.

KEYWORDS

Multi-Answer Open-Domain Question Answering, Check-Worthy Claim Detection, Natural Language Inference, Social Media.

SIGIR '23, July 23–27, 2023, Taipei, Taiwan

ACM ISBN 978-1-4503-9408-6/23/07…$15.00
https://doi.org/10.1145/3539618.3591907

Question: What can spread COVID-19?

Answer1: shoes

Supporting tweet: Please stop running outdoors. Your running shoes could have coronavirus. There's a global pandemic. Stop the spread.

Refuting tweet: Fake news alert - "shoes carry virus" - No, infected people that spread virus wearing shoes do. Are you going to drink my muddy footprints? Madness in ignorance and stupidity abounds. I will exercise totally alone with no risk to anyone. Please understand

Answer2: swimming water

Supporting tweet: Can Covid 19 virus spread through water? Seems so. Aisa hai toh will you wear masks while dancing in the Mumbai rains or while cooling off in a swimming pool? The latter can be dangerous, mind you.

Refuting tweet: Swimming Pool Water Unlikely to Spread Coronavirus But Facility Environments Need Careful Handling, Says Expert - Swimming World News

Figure 1: A TweetMythQA example. Note this entity question has 22 distinct answers annotated in the TweetMythQA. Two of the answers are listed here for demonstration purposes. MythQA aims to answer a factoid question with all distinct plausible answers and find contradictory stance evidence for each answer from a large corpus of tweets when available.

ACM Reference Format:
Yang Bai, Anthony Colas, and Daisy Zhe Wang. 2023. MythQA: Query-Based Large-Scale Check-Worthy Claim Detection through Multi-Answer Open-Domain Question Answering. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '23), July 23–27, 2023, Taipei, Taiwan.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3539618.3591907

1 INTRODUCTION

This paper proposes a new open-domain question answering (QA) task, MythQA, to tackle query-based large-scale check-worthy claim detection in an open-domain setting. Traditionally, the check-worthy claim detection module is used to determine which sentences in a given input text should be prioritized for fact-checking [12, 13]. Figure 2 shows the general steps of such a pipeline [33].

However, this setting is not efficient enough, especially when users are looking for distinct check-worthy claims on specific topics. It requires text snippets as input, which means that users still

have to collect relevant passages of concerned topics through other methods for the system to check. Consequently, it is not suitable for detecting check-worthy claims on a large scale. What's more, the traditional setting does not provide evidence for the predictions, which reduces its interpretability and requires an extra step (e.g., supporting evidence retrieval) before the final step of fact-checking.

A key characteristic of open-domain question answering (QA) is its use of external information sources, such as Wikipedia, to answer users' factoid questions. Recently, multi-answer open-domain QA has been proposed by [27]. One of its abilities is to help users find equally plausible distinct answers to users' queries. To introduce this ability in the check-worthy claim detection task, we propose MythQA(Query-based Large-scale Check-Worthy Claim Detection through Multi-Answer Open-domain Question Answering), the first open-domain QA-based large-scale distinct check-worthy claim detection framework which involves multi-answer open-domain question answering and contradictory stance mining. Specifically, the system must (1) find a set of distinct answers to the question, and (2) provide both supporting evidence and refuting evidence for each answer when available. The assumption is that any answer with contradictory stance evidence found may be misinformation that needs to be verified. New evaluation metrics are introduced in section 3.

To support the study of this task, we construct a dataset called TweetMythQA with 522 multi-answer questions manually created. All questions are annotated with multiple answers. Moreover, we collect relevant tweets for each distinct answer, then classify them into three categories: "Supporting", "Refuting", and "Neutral". In total, we annotated 5.3K tweets. We make sure every answer is accompanied by contradictory evidence (both supporting and refuting evidence). Two types of questions are covered by TweetMythQA, according to [15]: (1) factoid entity questions and (2) factoid yes/no questions. In the rest of the paper, we omit the term "factoid" when referring to the two types of questions. Details about data collection are provided in section 4.

We build the dataset using tweets because compared to other information sources such as government websites, Wikipedia, and news articles, Twitter contains more contradictory claims on many topics due to relatively weaker censorship. This leads to tweets spreading misinformation more rapidly, usually resulting in more severe consequences [1, 16, 35, 40]. Furthermore, as far as we are aware, there is no open-domain QA dataset based on social media. Our dataset presents distinct challenges compared to existing open-domain QA systems.

We provide benchmark results for existing zero-shot NLP models for this task. First, we examine the ability of information retrieval models to retrieve relevant tweets that include all distinct answers to a specific question. Second, we evaluate machine reading comprehension models on their ability to predict all distinct answers to the question given a set of relevant tweets. Third, following prior tasks [16, 36], we evaluate NLI models on misinformation (a.k.a stance detection) by equating the class labels, Supporting, Refuting, and Neutral to Entailment, Contradiction, and Neutral, respectively. Finally, an end-to-end evaluation is applied over a newly proposed pipeline system for MythQA consisting of the above NLP modules. New evaluation metrics are proposed to take multiple answers and contradictory stance evidence into consideration. Our results show that there is significant room for future work on this task.

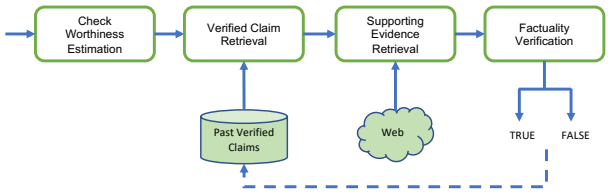


Figure 2: A traditional fact-checking pipeline.

The main contributions of our work can be summarized as follows:

(1) We introduce MythQA, a new task that extends current multi-answer open-domain QA to include contradictory stance evidence mining for each distinct answer in order to detect check-worthy claims at a large scale.
(2) We construct TweetMythQA, a dataset with 522 multi-answer open-domain questions on multiple controversial topics; 5.3K relevant stance evidence in total is collected from Twitter for the answers.
(3) We introduce baseline systems for MythQA and evaluate existing NLP systems for each component of the system using the TweetMythQA dataset, providing initial benchmarks and identifying key challenges for future models to improve upon.

2 RELATED WORKS

2.1 Open-domain Question Answering

Open-domain question answering(QA) aims to answer a factoid question in natural language by searching for evidence from a large-scale external information source such as Wikipedia [8, 37].

Recently, many benchmarks have been contracted to help the community better understand this task, such as CuratedTREC [6], WebQuestions [7], WikiMovies [26], and Qusar-T [10]. Several well-known machine reading comprehension(MRC) datasets were also curated to serve as open-domain QA benchmarks, such as OpenSQuAD [8], curated from SQuAD [31]; OpenTriviaQA [22], curated from TriviaQA [18]; OpenNaturalQuestions [22], curated from NaturalQuestions [20]. All of the benchmarks above use English Wikipedia as the external information source and all of them only require a single valid answer.

To study the ambiguity in open-domain questions, [27] proposed the first multi-answer open-domain QA benchmark, AmbigQA. It requires a system to find all equally plausible answers to an ambiguous question and generate a more specific question for each distinct answer.

However, in open-domain questions, ambiguity is not the only cause for multiple answers. It is common to encounter contradictory answers and parallel answers in our daily lives, and in many cases, they are exactly what users seek. Figure 1 shows examples of these cases, where "shoes" and "swimming water" are both plausible answers to the question "What can spread COVID-19?". By asking this question, users are likely seeking to find out all possible

things that can be used as a COVID-19 medium, so they can avoid them all. Additionally, the "Supporting" and "Refuting" claims indicate that there are contradictory answers to questions such as "Can shoes spread COVID-19?" and "Can swimming water spread COVID-19?". These contradictory answers are valuable for users and fact-checking organizations to be aware of the potential risks. Hence, one of the main motivations for this study is to explore the underexplored challenges of open-domain quality assurance that arise when answers are contradictory and parallel. To our knowledge, MythQA is the first study of its kind.

In contrast to previous works that used Wikipedia as their external information source, MythQA focuses on social media (Twitter). Compared to Wikipedia, social media has a significantly different data distribution as a result of the noise and informal nature of the text, and diverse (contradictory and parallel) claims about controversial topics. The features of social media pose new challenges for existing open-domain QA systems, but they provide a perfect corpus for MythQA, which aims to find check-worthy claims.

Lastly, MythQA is the first open-domain multi-answer QA benchmark with an emphasis on yes/no questions. AmbigNQ (the dataset proposed along with AmbigQA), in total, has only five yes/no questions in their development and training set, but none are multi-answer questions. TweetMythQA, on the other hand, contains 408 yes/no questions that are all multiple-answer questions.

2.2 Check-worthy Claim Detection

Check-worthy claim detection(CWCD) aims at predicting which sentences in an input text should be prioritized for fact-checking. The first work targeting the detection of check-worthy claims was the ClaimBuster system proposed by [12, 13]. The model is trained on manually annotated data where each sentence was marked as check-worthy factual, non-factual, or unimportant factual. The data consisted of transcripts of historical US election debates. Later, a larger version of the ClaimBuster dataset was published [3]. ClaimRank [17] is another important system for detecting check-worthy claims. It extended ClaimBuster to a new language(Arabic) and paid special attention to the context of each sentence. Other well-known works focused on political debates including [4, 5, 11, 29].

Recently, new domains are introduced for check-worthy claim detection, such as social media [28, 34], news articles [2], Wikipedia [39] and COVID-19 [1].

However, all previous works have focused on the setting of classification or ranking over sentences within a given text. As an alternative, MythQA aims to retrieve check-worthy claims directly from a large external information source (Twitter) based on users' queries.

3 TASK: MYTHQA

3.1 Task Setup

In this section, we first describe the MythQA task setup for entity questions and yes/no questions, respectively. Then, we describe the task setups for each subtask individually.

For entity questions, given a question q, the task is to predict a set of (answer, supporting evidence, refuting evidence) tuples: $\{(a_1, \mathcal{S}_1, \mathcal{R}_1), (a_2, \mathcal{S}_2, \mathcal{R}_2), ..., (a_n, \mathcal{S}_n, \mathcal{R}_n)\}$, where each a_i is a plausible answer to q, each $\mathcal{S}_i$ is a set of supporting evidence tweet of a_i, and each $\mathcal{R}_i$ is a set of refuting evidence tweet of a_i. Note that to be more practical and increase the chance of finding gold evidence, we predict a set of supporting/refuting evidence instead of a single one.

For yes/no questions, since the contradictory evidence of answers "YES" and "NO" are symmetric, hence, a system is only required to find supporting evidence for each answer, respectfully. In particular, given a question q, predict a set of (answer, supporting evidence) pairs: $\{(a_1, \mathcal{S}_1), (a_2, \mathcal{R}_2)\}$.

Tweet retrieval
Given a question q or a claim c, output top-k relevant tweets, $T = \{t_1, t_2, ..., t_k\}$ from a large tweet corpus C, where k is a given hyperparameter.

Stance Detection
Given an (c, t) pair, predict their stance relation from Supporting, Refuting, and Neutral.

Multiple Answer Prediction
Given a question q and the top-k relevant tweets T, output a set of distinct plausible answers $A = \{a_1, a_2, ..., a_n\}$, where n is unknown.

Contradictory Stance Mining
Given a claim c, return the top-e supporting tweet evidence, and the top-e refuting tweet evidence from a large tweet corpus C when available, where e is a given hyperparameter. This task could be thought of as a combination of relevant claim retrieval and stance detection.

A claim c can be constructed automatically as simply as by concatenating a distinct answer and its corresponding question, e.g, "What can spread COVID-19? Answer is swim water.", where " Answer is " serves as a connection string.

3.2 Evaluation Metrics

To evaluate model performance, we present several ways to compare a model prediction with m (answer, supporting evidence, refuting evidence) tuples $(a_1, \mathcal{S}_1, \mathcal{R}_1), \ldots, (a_m, \mathcal{S}_m, \mathcal{R}_m)$ with a gold reference set with n tuples $(\bar{a}_1, \bar{\mathcal{S}}_1, \bar{\mathcal{R}}_1), \ldots, (\bar{a}_n, \bar{\mathcal{S}}_n, \bar{\mathcal{R}}_n)$, where $size(\mathcal{S}_i) = size(\mathcal{R}_i) = e$.

We assign each predicted (answer, supporting evidence, refuting evidence) tuple $(a_i, \mathcal{S}_i, \mathcal{R}_i)$ a *supporting evidence match score* se_i and a *refuting evidence match score* re_i based on if any gold evidence is included. We use exact match when comparing the predicted answer/evidence and the gold answer/evidence.

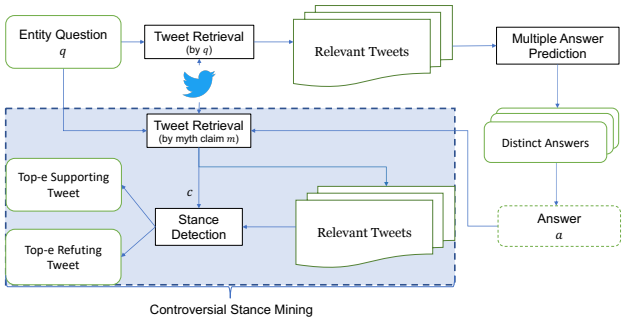
$$se_i = \max_{1 \le j \le n} \mathbb{I}[a_i = \bar{a}_j] f(\mathcal{S}_i, \bar{\mathcal{S}}_j) \tag{1}$$

$$re_i = \max_{1 \le j \le n} \mathbb{I}[a_i = \bar{a}_j] f(\mathcal{R}_i, \bar{\mathcal{R}}_j) \tag{2}$$

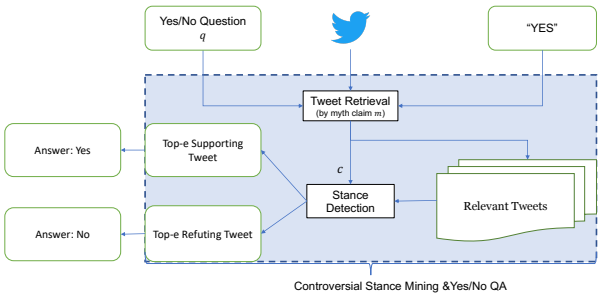
where f is defined as:

$$f(E, \bar{E}) = \mathbb{I}[E \cap \bar{E} \neq \emptyset] \tag{3}$$

where E is a set of evidence text.



(a) MythQA pipeline for entity questions. A claim m is composed of a question q and a distinct answer a.



(b) MythQA pipeline for yes/no questions. A claim m is composed of a question q and a positive answer "YES".

Figure 3: MythQA pipeline.

The correctness score c_i of answer a_i is the average of se_i and re_i:

$$c_i = (se_i + re_i)/2 \tag{4}$$

Note that, for a yes/no question, finding correct supporting evidence for the NO answer is equivalent to finding correct refuting evidence for the YES answer, and vice versa, hence, c_i could be simplified to:

$$c_i = se_i \tag{5}$$

We calculate F1 treating the c_i as measures of correctness:

$$\text{prec}_f = \frac{\sum_i c_i}{m} \tag{6}$$

$$\text{rec}_f = \frac{\sum_i c_i}{n} \tag{7}$$

$$\text{F1}_f = \frac{2 \times \text{prec}_f \times \text{rec}_f}{\text{prec}_f + \text{rec}_f} \tag{8}$$

We consider two choices for F1_f: $\text{F1}_{\textbf{ans}}$, and $\text{F1}_{\textbf{CONTRO}}@e$. $\text{F1}_{\textbf{ans}}$ is the F1 score on answers only, where f always yields 1. $\text{F1}_{\textbf{CONTRO}}@e$ further checks if any gold evidence is included in the top-e evidence predictions, where f is defined in equation (3).

To evaluate a retriever's result with respect to the number of distinct answers found in the top-k retrieval, we propose $MHits@k$:

$$MHits@k = Hits@k \times \frac{\sum_i \mathbb{I}[\bar{s}_i \in T \ \vee \ \bar{r}_i \in T]}{n} \tag{9}$$

For failing to retrieve relevant evidence for all distinct answers to the question, the retriever will be penalized.

4 DATA COLLECTION

We construct TweetMythQA examples in two phases: multi-answer QA pair generation and stance evidence collection. An overview of the annotation process is shown in Figure 4. We build a large external information corpus using cleaned relevant tweets collected during annotation using the Twitter API. We ensure that (1) all the tweets in the corpus are relevant to the topics of the questions in TweetMythQA; (2) all the tweet ids in TweetMythQA are included in the Corpus; (3) retweets and duplicate tweets are removed. In total, around 200K English tweets were collected as an external information source. For training and quality control, high-quality annotators are carefully selected and hired from three master students in the computer science department. Annotations are sampled and checked by co-authors during the process to ensure they are high-quality.

Q Type	# Q	# Avg.SupEve/Ans	# Avg.RefEve/Ans
Entity	114	2.89	2.47
Yes/No	408	2.92	2.49
Overall	522	2.91	2.49

Table 1: Data statistics. In TweetMythQA, all questions have multiple answers. All answers have multiple pieces of contradictory evidence.

	# Ans / Q				
Question Type	1	2	3	4+	# Avg.Ans/Q
Entity	0	41	35	38	3.43
Yes/No	0	408	0	0	2

Table 2: The number of answers per question.

Stance Label	Count	Percentage
Supporting	2318	43.88%
Refuting	1980	37.49%
Neutral	984	18.63%

Table 3: Distribution of evidence by stance.

4.1 Annotation Process

Multi-Answer QA Pair Generation

The process of creating a question in TweetMythQA involves three major steps: selecting a controversial topic, searching for relevant claims, and creating the question. In particular, to create an entity question, annotators must summarize claims from the same topic,

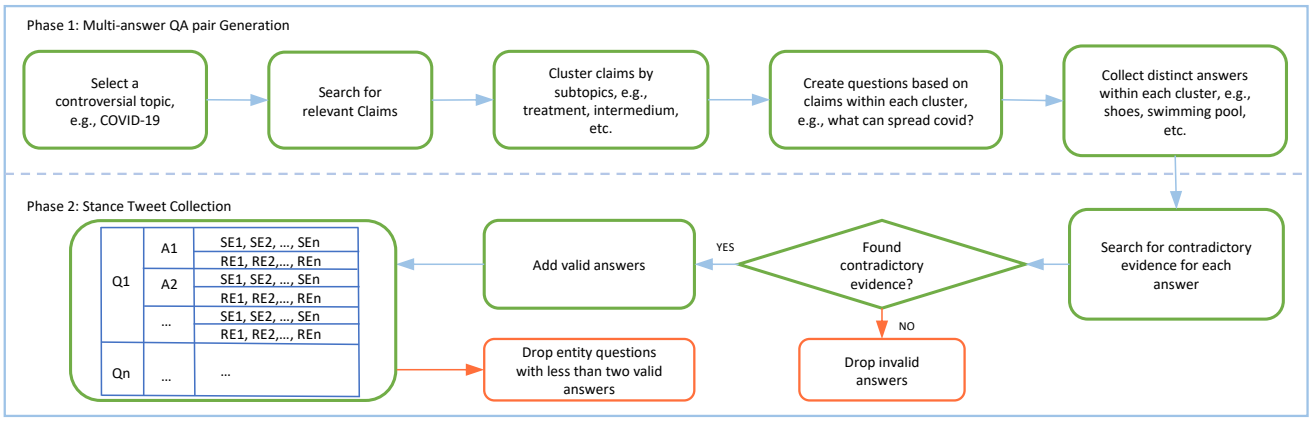


Figure 4: Annotation process of TweetMythQA. The terms Q, A, SE, and RE refer to the question, answer, supporting evidence, and refuting evidence, respectively.

while yes/no questions are created by directly paraphrasing claims from the selected topics into verification questions[1].

The social media surrounding controversial topics tends to contain many contradictory claims. Thus, when generating questions, a controversial topic (e.g., COVID-19) is selected from a list of controversial topics on Wikipedia[2].

Then, relevant claims are gathered through myth-busting and fact-checking websites, such as WHO[3], FEMA[4], Wikipedia[5], NewsGuard[6], Google Fact Check Tools[7], EUvsDisinfo[8], Polygraph.info[9], PolitFact[10], etc

After retrieving the claims, they are manually clustered based on different aspects/subtopics of the controversial topic, such as medium, prevention, source, etc.

By summarizing the claims in each cluster, the annotator formulates an entity question that can be answered by any of the distinct claims within the cluster. For example, based on claims of the same subtopic cluster: "The likelihood of shoes spreading COVID-19 is very low", "Water or swimming does not transmit the COVID-19 virus", etc. A question such as: "What can spread COVID-19?" could be created.

Finally, short answers are derived from the claims, for example, shoes, swimming water, etc. Following [22], all annotated answers are shorter than 5 tokens.

[1]Noted that the claims used to create entity questions can also be used to create yes/no questions, resulting in the fact that many yes/no questions in our dataset have the same format as entity questions. For example, "What can spread COVID-19?" and "Can shoes spread COVID-19?". We allow such overlap in the dataset because we evaluate entity questions and yes/no questions separately.
[2]https://en.wikipedia.org/wiki/Wikipedia:List_of_controversial_issues
[3]https://www.who.int/emergencies/diseases/novel-coronavirus-2019/advice-for-public/myth-busters
[4]https://www.fema.gov/disasters/coronavirus/rumor-control
[5]https://en.wikipedia.org/wiki/COVID-19_misinformation
[6]https://www.newsguardtech.com
[7]https://toolbox.google.com/factcheck/explorer
[8]https://euvsdisinfo.eu/disinformation-cases/
[9]https://www.polygraph.info/
[10]https://www.politifact.com/

Stance Tweets Collection

Each distinct answer and corresponding question make up a distinct claim. In the second phase, we collect contradictory stance evidence for every distinct claim through the Twitter API. Annotators are rewarded for finding both supporting and refuting evidence for each answer.

We observe that the results directly returned by the Twitter API are very noisy, which makes it hard to find relevant tweets that can be used as stance evidence for most of the claims collected from the first-phase annotation, let alone finding contradictory evidence.

To reduce the annotation difficulty, we developed a heuristic tool that can help annotators search through the Twitter API and suggests the top 100 potentially controversial tweets to annotators. In particular, it can automatically construct multiple queries for each claim based on a template and a list of alias of the topic entity given by the annotators. For example, given a template: "shoes can spread TOPIC_ENTITY", and a list of alias of the TOPIC_ENTITY, e.g., *COVID-19, covid, Coronavirus, Coronavirus 2, SARS-CoV-2, novel coronavirus-2019, Wuhan virus, etc.* Queries of the same template with different alias of the TOPIC_ENTITY will be constructed and sent to the Twitter API. We find this practice can significantly increase the chance of finding contradictory stance evidence.

For the raw tweets returned by the Twitter API, the tool does a series of data cleaning such as: dropping retweets and tweets with similar content, etc. After that, a state-of-the-art pretrained dense passage ranker, DPR (Reference), is applied to retrieve the top-1000 tweets based on the semantic similarity between the query and the cleaned tweets. Then the top-1000 tweets are clustered in 5 clusters using the K-means algorithm based on their embeddings generated by the indexer that is used by the DPR. Finally, 100 tweets consisting of the top-20 tweets of each cluster are returned as the final set for the annotators to check.

4.2 Dataset Statistics

In TweetMythQA, two types of factoid questions are annotated: entity questions and yes/no questions. Table 1 shows that in TweetMythQA, all questions are multi-answer questions, and each answer

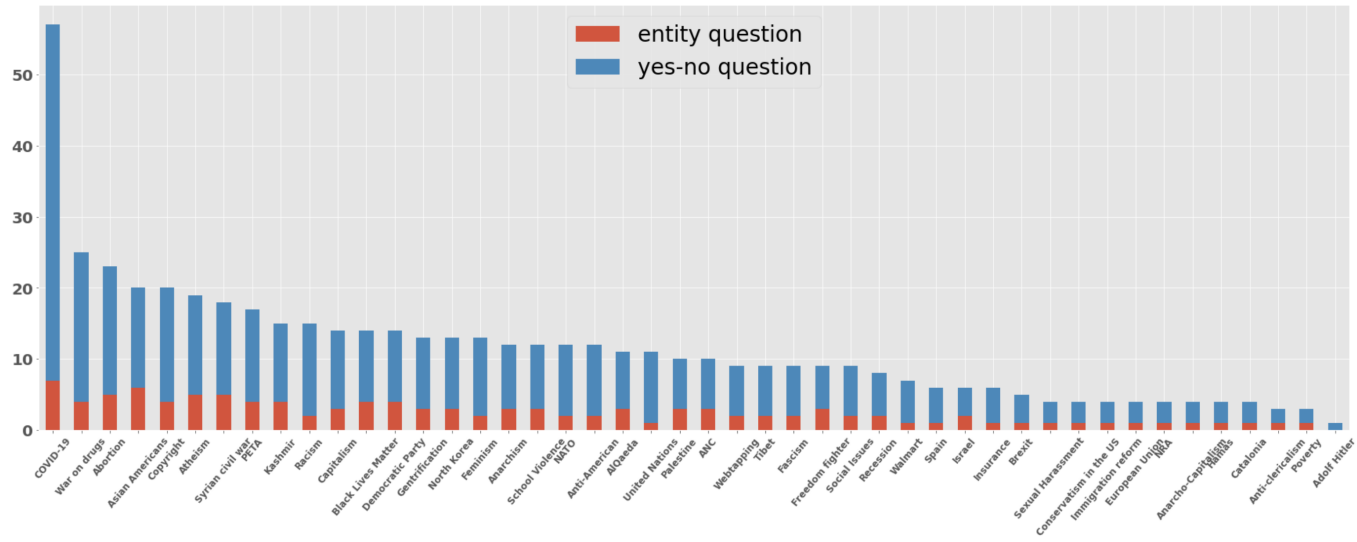


Figure 5: Distribution of topics and question types in TweetMythQA.

is backed up with multiple pieces of contradictory evidence(both supporting and refuting evidence). Table 2 gives more detailed statistics about the number of distinct answers per question. Each entity question has an average of 3.5 distinct answers, and all yes/no questions have an average of two distinct responses. Table 3 shows the distribution of stance evidence. In total, there are 44 topics covered by TweetMythQA.

In total, there are 44 topics covered by TweetMythQA. Figure 5 shows the distribution of each type of question among topics in TweetMythQA. By the category of the controversial topic list on Wikipedia, besides COVID-19 is from the "Science, biology, and health" section, all the rest topics in the TweetMythQA are selected from the section "Politics and economics" due to limited annotation time. More topics in other sections will be covered in our future work.

4.3 Quality Control

Due to the significant portion of search and summary procedures involved, and often the requirement of strong background knowledge for some topics, this annotation task is significantly more difficult than constructing other single-answer, single-stance-evidence QA datasets, and normal stance detection datasets. We sample and check annotations frequently during the annotation process in order to ensure quality. In addition, we use handy communication tools such as Slack[11] and Zoom[12] to answer any technical issues as soon as possible. Prior to annotating the remaining examples, any disagreements were discussed and resolved.

Furthermore, we ensure that each question has multiple answers, and each answer has both supporting and refuting evidence. As a result, there is a high annotation rejection rate, which results in a high-quality dataset at a high cost.

[11]https://slack.com/
[12]https://zoom.us/

4.4 More Details on Annotation Work

The three annotators are all first-year graduate students in the computer science department. All of them are male.

Annotators are paid 13 dollars/hour with certain expectations and rewards for different phases of annotation.

Phase-one annotations require each annotator to complete four entity questions per hour and provide at least two distinct answers to each question. More distinct answers to each entity question are encouraged. There will be a bonus of $3.25 for every additional four distinct answers.

Phase-two annotation requires each annotator to collect 120 pieces of supporting/refuting evidence per hour for the answers collected in the last annotation phase. Since it is difficult to find both supporting and refuting evidence, annotators are rewarded with the full hourly payment if they can find two pairs of non-overlapping supporting-refuting evidence for an answer, i.e., two distinct supporting evidence and two distinct refuting evidence.

5 PERFORMANCE OF BENCHMARK MODELS

In real life, claims generate and evolve rapidly. Annotating a dataset large enough for training would be very expensive and would never keep up with the speed at which data changes in the field of check-worthy claim detection (CWCD). Thus, it is desirable that CWCD systems be data-efficient, i.e., trained with little or no supervision.

In this section, we investigate various zero-shot baselines for different sub-tasks within our MythQA task outlined in Section 3. We test a diverse set of baselines in an evaluation-only setting for the sub-tasks below.

5.1 Tweet retrieval

We evaluate a sparse retrieval approach and a dense retrieval approach for the tweet retrieval sub-task. Specifically, for the sparse

retrieval approach, we choose BM25 [32], a famous traditional scoring algorithm using bag-of-words representations. For the dense retrieval approach, we choose DPR[13][14] [19], a state-of-the-art on NQ-open, which does nearest-neighbor search on transformer-encoded representations. We use the Pyserini IR toolkit [24] to implement both BM25 and DPR tweet retrievers.

Results We present the performance of two tweet retrievers in Table 4. For entity questions and yes/no questions, we see that BM25 performs better than pretrained DPR. This advantage is more significant for yes/no questions especially when the retrieval number is small. Besides, we find that *MHit* scores are clearly lower than the *Hit* scores for the entity questions. This indicates the challenges in multi-answer retrieval, i.e., maximizing the coverage of relevant evidence(supporting and refuting evidence) of distinct answers in the top-k retrievals.

5.2 Stance Detection

Following [16, 42], we leverage NLI models for the zero-shot stance detection task. In particular, We evaluate existing NLI models: BERT-large [9][15], ALBERT-large [21][16], XLNet-large [41][17], BART-large [23][18], Roberta-large [25][19], and DeBERTa [14][20]. All models are pretrained over MNLI [38]. We construct our stance detection evaluation dataset, TweetMythSD, which consists of claim-evidence pairs in the TweetMythQA. Each claim is composed of a distinct answer and the corresponding question as described in section 3. Table 3 shows statistics of the TweetMythSD.

Results: The results of the stance detection are shown in Table 5. DeBERTa-large achieves the highest F1 for the Supporting and Refuting class, as well as the highest macro-averaged Procession, Recall, and F1 for all the pretrained models we evaluated. On the other hand, most models do not perform well on the Neutral class. In addition, we observe that Refuting and Neutral results differ greatly while Supporting results are relatively stable. The reason may be that, in our dataset, there is a greater data shift in refuting and neutral examples than in supporting examples. Different results between the models can be attributed to differing generalization abilities for each stance class.

5.3 Multiple Answers Prediction

Due to the difference between the nature of entity questions and yes/no questions, we propose a specific pipeline for each of the two types of questions, as shown in Figure 3.

Entity Questions: To predict distinct answers to entity questions, we use a pretrained machine reading comprehension(MRC) model. In particular, we evaluate an extractive MRC model DPR reader[21] [19] pretrained on Natural Questions [20], and a generative MRC model T5[22] [30] pretrained on SQuAD [31].

Neither the original DPR reader nor the T5 reader was designed to predict multiple answers. For this task, we propose a simple but efficient method for predicting multiple answers. Rather than train a model to decide the number of answers, n, to predict [27], we make n a hyperparameter, i.e., an MRC model only needs to find the top-n distinct answers from the top-k retrieved tweets, where $k >> n$. For each tweet, the MRC model only needs to predict one answer. Candidate answers are ranked based on a weighted average of tweet retrieval scores from the retriever model and answer span scores from the MRC model.[23]. A normalization process will also be applied to the answers. Answers that are duplicated will be removed. By doing so, we can make direct use of MRC models pretrained on other large datasets of single-answer MRCs.

Yes/No Questions: Rather than using a machine reading comprehension (MRC) model to predict different yes/no answers, we directly do contradictory stance mining for the positive claims. Each positive claim is composed of a yes/no question and the "YES" answer as described in section 3. Findings of supporting and refuting evidence of the positive claim, indicate the answers "YES" and "NO" to the yes/no question respectively. If no relevant evidence is found, output "NOT SURE". On the basis of the logits of the stance detection model, the top-ranked evidence of "YES" and "NO" answers is selected.

Both intrinsic and extrinsic evaluations are done for entity and yes/no questions.

Intrinsic: The prediction is conditioned on a question and relevant tweets that are annotated for the question.

Extrinsic: The prediction is conditioned on a question and relevant tweets retrieved by the tweet retriever from the whole corpus.

Results: Table 6 and Table 7 show the performance on multiple answer prediction for entity questions and yes/no questions, respectively. As Table 6 presents, the pretrained DPR Reader (on NQ) performs marginally better than the pretrained T5 (on SQuAD) in extrinsic evaluation. Nonetheless, neither of the pretrained models perform well on this task, indicating there is huge room for improvement in multi-answer open-domain QA on social media. Promising next steps include (1) domain adaptive training over social media data; (2) using better answer normalization and selection rules to distinguish distinct answers from the same answers in different presentations even from the ones with typos which is quite common in social media, e.g., 'the USA', 'the United States', 'America', etc. The difference between the intrinsic and extrinsic evaluation scores shows how much the performance of the tweet retriever affects the final results.

Table 7 shows how retrieval number(k) affects the multiple answer prediction for yes/no questions. In this experiment, we use the best-performed tweet retriever and stance detection model in our previous experiments, i.e., BM25 + DeBERTa-large. We can see that, for intrinsic evaluation, the $F1_{ans}$ does not change much when k is greater than 10. This is because, on average, there is only 5.4 relevant stance evidence annotated for each yes/no question. In extrinsic evaluation, we observe that, when k is increased to 100, the $F1_{ans}$ of

[13]https://huggingface.co/facebook/dpr-question_encoder-multiset-base
[14]https://huggingface.co/facebook/dpr-ctx_encoder-multiset-base
[15]https://huggingface.co/madlag/bert-large-uncased-mnli
[16]https://huggingface.co/anirudh21/albert-large-v2-finetuned-mnli
[17]https://huggingface.co/ynie/xlnet-large-cased-snli_mnli_fever_anli_R1_R2_R3-nli
[18]https://huggingface.co/facebook/bart-large-mnli
[19]https://huggingface.co/roberta-large-mnli
[20]https://huggingface.co/microsoft/deberta-large-mnli
[21]https://huggingface.co/facebook/dpr-reader-single-nq-base
[22]https://huggingface.co/valhalla/t5-base-qa-qg-hl

[23]For generative MRC model such as T5, we only use the tweet retrieval score to rank the answer.

	BM25				DPR			
Question Type	MH@100	H@100	MH@1K	H@1K	MH@100	H@100	MH@1K	H@1K
Entity	79.67	92.98	**96.13**	99.12	70.29	88.60	**92.90**	98.25
Yes/No	97.06	97.06	**99.26**	99.26	80.88	80.88	**95.10**	95.10
Overall	93.26	96.17	**98.58**	99.23	78.57	82.57	**94.62**	95.79

Table 4: Tweet Retrieval Performance. We present evaluation results for relevant tweet retrieval (i.e., tweets that either support or refute one or more distinct answers). *MH@k* is the abbreviation of *MHit@k*, a new metric that we introduce in section 3.2. It is important to note that for yes/no questions, *Hit@k* is equivalent to *MHit@k* because it either finds relevant tweets for all answers ("YES" and "NO") or for no answers.

	Macro Avg.			Supporting			Refuting			Neutral		
Model	P	R	F	P	R	F	P	R	F	P	R	F
BERT-large	23.95	35.47	27.63	52.68	76.79	62.49	3.70	0.05	0.10	15.47	**29.58**	**20.31**
ALBERT-large	25.95	35.49	26.96	52.43	**90.38**	66.36	12.82	0.25	0.50	12.59	15.83	14.03
XLNet-large	30.20	33.05	27.50	49.26	81.67	61.45	30.40	5.80	9.75	10.94	11.67	11.29
BART-large	41.90	43.58	39.67	52.87	73.94	61.65	56.16	56.16	56.16	16.67	0.63	1.20
RoBERTa-large	48.21	47.62	43.90	54.98	85.76	67.00	70.24	55.75	62.16	19.40	1.35	2.53
DeBERTa-large	**50.12**	**48.90**	**44.74**	**55.83**	89.00	**68.62**	**73.85**	**57.08**	**64.39**	**20.69**	0.63	1.21

Table 5: Stance Detection Performance. We present evaluation results for classifying claim-tweet pairs into Supporting, Refuting, and Neutral classes. The precision (P), recall (R), and F1-score (F1) are presented for each class, as well as macro averaged values. All models are pretrained on MNLI.

	Intrinsic ($F1_{ans}$)			Extrinsic ($F1_{ans}$)		
Models	m=1	m=5	m=10	m=1	m=5	m=10
			k = 10			
DPR Reader	16.10	26.56	26.94	14.59	20.68	19.84
T5	16.45	25.45	27.06	12.17	20.20	19.69
			k = 100			
DPR Reader	16.51	27.88	29.69	15.10	**20.83**	16.24
T5	16.45	25.84	**29.88**	12.17	19.60	16.21
			k = 1000			
DPR Reader	16.51	27.88	29.69	15.10	**20.83**	16.24
T5	16.45	32.92	**29.88**	12.17	19.60	16.21

Table 6: Multiple Answer Prediction Performance for Entity Questions. In extrinsic evaluation, BM25 is used to retrieve tweets. *k* refers to, for each question, the number of tweets that are retrieved as context. *m* is a hyperparameter that indicates the number of answers the system needs to predict.

	Intrinsic		Extrinsic(E2E)	
k	$F1_{ans}$	$F1_{CONTRO}$@1	$F1_{ans}$	$F1_{CONTRO}$@1
5	89.71	54.58	78.02	17.77
10	91.83	50.98	91.09	19.04
100	91.99	49.80	99.67	2.70
500	91.99	49.80	100	0.49
1000	91.99	49.80	100	0.25

Table 7: Multiple Answer Prediction Performance for Yes/No Questions. We use the best-performed tweet retriever and stance detection model in our previous experiments, i.e., BM25 + DeBERTa-large. *k* refers to, for each claim, the number of tweets that are retrieved for stance detection. E2E refers to end-to-end.

the model increases to close to 100, while $F1_{CONTRO}$@1 drops significantly to 2.70.

Several insights can be drawn from this: (1) Unlike multi-answer predictions for entity questions, $F1_{ans}$ itself is insufficient to reflect the actual performance of the multi-answer prediction task for yes/no questions. The reason is that no matter how poor the retrieval or stance detector's performance may be, $F1_{ans}$ will always be close to 100 when the size of the retrieval is large enough. (2) However, the $F1_{CONTRO}$@*e* scores are indicative of the actual performance of the multi-answer prediction for the yes/no questions, since they compare the predicted evidence with the gold evidence. (3) In addition, we observe that the larger the retrieval number the worse the $F1_{CONTRO}$@1 score of the multi-answer prediction for the yes/no question. This reflects how the performance of the stance detection model detracts from the overall performance of the multi-answer question predictions because the top evidence is selected based on the logits of the stance detection model. (4) Compared to the intrinsic setting, the drop in extrinsic performance illustrates the challenges within the tweet retrieval model.

	Intrinsic				Extrinsic(E2E)			
	$F1_{ans}$	$F1_{CONTRO}@e$			$F1_{ans}$	$F1_{CONTRO}@e$		
Q Type		e=1	e=10	e=100		e=1	e=10	e=100
Entity	100	5.37	22.84	58.90	20.83	0.71	4.34	**12.48**
Yes/No	91.99	49.80	64.30	64.30	100	0.25	4.78	**29.04**

Table 8: Stance Mining and End-to-end MythQA Performance. Experiment setup: Retriever: BM25, retrieval number: 1000; machine reader: DPR Reader(on NQ), number of predicted answers: 5; stance detector: DeBERTa-large(on MNLI). Q refers to the question. *e* refers to the number of stance(supporting/refuting) evidence included in the prediction. E2E refers to end-to-end.

5.4 Contradictory Stance Mining

Contradictory stance mining aims to find contradictory(both supporting and refuting) claims for a given claim as described in section 3.

For entity questions, we perform both intrinsic and extrinsic evaluations.

Intrinsic: Input claims are composed of gold answers and corresponding questions.

Extrinsic: Input claims are composed of the extrinsic MRC outputs and the corresponding questions. This is equivalent to an end-to-end MythQA evaluation.

For yes/questions, the experiment setting of contradictory stance mining is equivalent to multiple answers prediction as described in section 5.3.

Results: The intrinsic and extrinsic (end-to-end MythQA) stance mining results are presented in Table 8. In comparison to intrinsic evaluation results, the huge performance drop in the extrinsic evaluation of entity questions indicates that multiple-answer prediction is the bottleneck of the entire pipeline. For yes/no questions, the analysis presented in section 5.3 is also applicable since, as noted, in this setting, multi-answer prediction for yes/no questions is equivalent to contradictory stance mining and follows the same metrics.

Overall performance is low both intrinsically and extrinsically, which highlights the difficulty of the task. This can be attributed to two factors. First, multiple modules are involved in the pipeline. Mistakes accumulate at each stage. Further improvements can be made to each module. Second, there is a lack of annotated data, especially for open-domain QA over social media; future work can explore how to maximize the use of supervision from other data collected from social media.

6 CONCLUSION & FUTURE WORK

We introduced MythQA, a new multi-answer open-domain question answering(QA) task that involves contradictory stance mining for query-based large-scale check-worthy claim detection. We constructed TweetMythQA, a dataset with 5.3K contradictory evidence annotations on 1.2K distinct answers to 522 manually generated multi-answer questions. Furthermore, we present a baseline system for MythQA and evaluate existing NLP models for each component using the TweetMythQA dataset. We also highlight the potential areas for improvement.

Future research on MythQA may include (1) domain-adaptive training over social media data, (2) improved answer normalization and selection rules to distinguish distinct answers from the same answers in different presentations, including those with typos often seen in social media, and (3) more carefully evaluating its effectiveness in downstream fact-checking systems.

ACKNOWLEDGMENTS

Our sincere thanks go out to the anonymous reviewers who took the time to provide constructive comments. This work is partially supported by DARPA under Award No. FA8750-18-2-0014 (AIDA/GAIA).

REFERENCES

[1] Firoj Alam, Shaden Shaar, Alex Nikolov, Hamdy Mubarak, Giovanni Da San Martino, Ahmed Abdelali, Fahim Dalvi, Nadir Durrani, Hassan Sajjad, Kareem Darwish, and Preslav Nakov. 2021. Fighting the COVID-19 Infodemic: Modeling the Perspective of Journalists, Fact-Checkers, Social Media Platforms, Policy Makers, and the Society. In *EMNLP*.

[2] Tariq Alhindi, Brennan McManus, and Smaranda Muresan. 2021. What to Fact-Check: Guiding Check-Worthy Information Detection in News Articles through Argumentative Discourse Structure. In *Proceedings of the 22nd Annual Meeting of the Special Interest Group on Discourse and Dialogue*. 380–391.

[3] Fatma Arslan, Naeemul Hassan, Chengkai Li, and Mark Tremayne. 2020. A benchmark dataset of check-worthy factual claims. In *Proceedings of the International AAAI Conference on Web and Social Media*, Vol. 14. 821–829.

[4] Pepa Atanasova, Lluís Màrquez i Villodre, Alberto Barrón-Cedeño, T. Elsayed, Reem Suwaileh, Wajdi Zaghouani, Spas Kyuchukov, Giovanni Da San Martino, and Preslav Nakov. 2018. Overview of the CLEF-2018 CheckThat! Lab on Automatic Identification and Verification of Political Claims. Task 1: Check-Worthiness. *ArXiv* abs/1808.05542 (2018).

[5] Pepa Atanasova, Preslav Nakov, Georgi Karadzhov, Mitra Mohtarami, and Giovanni Da San Martino. 2019. Overview of the CLEF-2019 CheckThat! Lab: Automatic Identification and Verification of Claims. Task 1: Check-Worthiness. In *CLEF*.

[6] P. Baudis and J. Sedivý. 2015. Modeling of the Question Answering Task in the YodaQA System. In *CLEF*.

[7] Jonathan Berant, Andrew K. Chou, Roy Frostig, and Percy Liang. 2013. Semantic Parsing on Freebase from Question-Answer Pairs. In *EMNLP*.

[8] Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading Wikipedia to Answer Open-Domain Questions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Vancouver, Canada, 1870–1879. https://doi.org/10.18653/v1/P17-1171

[9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *ArXiv* abs/1810.04805 (2019).

[10] Bhuwan Dhingra, Kathryn Mazaitis, and William W. Cohen. 2017. Quasar: Datasets for Question Answering by Search and Reading. *ArXiv* abs/1707.03904 (2017).

[11] Pepa Gencheva, Preslav Nakov, Lluís Màrquez i Villodre, Alberto Barrón-Cedeño, and Ivan Koychev. 2017. A Context-Aware Approach for Detecting Worth-Checking Claims in Political Debates. In *RANLP*.

[12] Naeemul Hassan, Bill Adair, James T Hamilton, Chengkai Li, Mark Tremayne, Jun Yang, and Cong Yu. 2015. The quest to automate fact-checking. In *Proceedings of the 2015 computation+ journalism symposium*.

[13] Naeemul Hassan, Fatma Arslan, Chengkai Li, and Mark Tremayne. 2017. 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*. 1803–1812.

[14] Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. DeBERTa: Decoding-enhanced BERT with Disentangled Attention. *ArXiv* abs/2006.03654 (2021).

[15] Wei He, Kai Liu, Jing Liu, Yajuan Lyu, Shiqi Zhao, Xinyan Xiao, Yuan Liu, Yizhong Wang, Hua Wu, Qiaoqiao She, Xuan Liu, Tian Wu, and Haifeng Wang. 2018. DuReader: a Chinese Machine Reading Comprehension Dataset from Real-world Applications. In *QA@ACL*.

[16] Tamanna Hossain, Robert L Logan IV, Arjuna Ugarte, Yoshitomo Matsubara, Sean D. Young, and Sameer Singh. 2020. COVIDLies: Detecting COVID-19

Misinformation on Social Media. In *NLP4COVID@EMNLP*.
[17] Israa Jaradat, Pepa Gencheva, Alberto Barrón-Cedeño, Lluís Màrquez i Villodre, and Preslav Nakov. 2018. ClaimRank: Detecting Check-Worthy Claims in Arabic and English. In *NAACL*.
[18] Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. 2017. TriviaQA: A Large Scale Distantly Supervised Challenge Dataset for Reading Comprehension. In *ACL*.
[19] Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Yu Wu, Sergey Edunov, Danqi Chen, and Wen tau Yih. 2020. Dense Passage Retrieval for Open-Domain Question Answering. *ArXiv* abs/2004.04906 (2020).
[20] T. Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur P. Parikh, Chris Alberti, D. Epstein, Illia Polosukhin, J. Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc V. Le, and Slav Petrov. 2019. Natural Questions: A Benchmark for Question Answering Research. *Transactions of the Association for Computational Linguistics* 7 (2019), 453–466.
[21] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. *ArXiv* abs/1909.11942 (2020).
[22] Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. 2019. Latent Retrieval for Weakly Supervised Open Domain Question Answering. In *ACL*.
[23] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In *ACL*.
[24] Jimmy J. Lin, Xueguang Ma, Sheng-Chieh Lin, Jheng-Hong Yang, Ronak Pradeep, Rodrigo Nogueira, and David R. Cheriton. 2021. Pyserini: A Python Toolkit for Reproducible Information Retrieval Research with Sparse and Dense Representations. *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval* (2021).
[25] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *ArXiv* abs/1907.11692 (2019).
[26] Alexander H. Miller, Adam Fisch, Jesse Dodge, Amir-Hossein Karimi, Antoine Bordes, and J. Weston. 2016. Key-Value Memory Networks for Directly Reading Documents. In *EMNLP*.
[27] Sewon Min, Julian Michael, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2020. AmbigQA: Answering Ambiguous Open-domain Questions. In *EMNLP*.
[28] Preslav Nakov, Giovanni Da San Martino, Tamer Elsayed, Alberto Barrón-Cedeño, Rubén Míguez, Shaden Shaar, Firoj Alam, Fatima Haouari, Maram Hasanain, Watheq Mansour, et al. 2021. Overview of the CLEF–2021 CheckThat! Lab on Detecting Check-Worthy Claims, Previously Fact-Checked Claims, and Fake News. In *International Conference of the Cross-Language Evaluation Forum for European Languages*. Springer, 264–291.
[29] Ayush Patwari, Dan Goldwasser, and Saurabh Bagchi. 2017. TATHYA: A Multi-Classifier System for Detecting Check-Worthy Statements in Political Debates. *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management* (2017).
[30] Colin Raffel, Noam M. Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, W. Li, and Peter J. Liu. 2020. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *ArXiv* abs/1910.10683 (2020).
[31] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ Questions for Machine Comprehension of Text. In *EMNLP*.
[32] Stephen E. Robertson and Hugo Zaragoza. 2009. The Probabilistic Relevance Framework: BM25 and Beyond. *Found. Trends Inf. Retr.* 3 (2009), 333–389.
[33] Shaden Shaar, Nikolay Babulkov, Giovanni Da San Martino, and Preslav Nakov. 2020. That is a Known Lie: Detecting Previously Fact-Checked Claims. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Online, 3607–3618. https://doi.org/10.18653/v1/2020.acl-main.332
[34] Shaden Shaar, Alex Nikolov, Nikolay Babulkov, Firoj Alam, Alberto Barrón-Cedeño, Tamer Elsayed, Maram Hasanain, Reem Suwaileh, Fatima Haouari, Giovanni Da San Martino, and Preslav Nakov. 2020. Overview of CheckThat! 2020 English: Automatic Identification and Verification of Claims in Social Media. In *CLEF*.
[35] Kai Shu, Amy Sliva, Suhang Wang, Jiliang Tang, and Huan Liu. 2017. Fake news detection on social media: A data mining perspective. *ACM SIGKDD explorations newsletter* 19, 1 (2017), 22–36.
[36] James Thorne, Andreas Vlachos, Oana Cocarascu, Christos Christodoulopoulos, and Arpit Mittal. 2018. The Fact Extraction and VERification (FEVER) Shared Task. In *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*. Association for Computational Linguistics, Brussels, Belgium, 1–9. https://doi.org/10.18653/v1/W18-5501
[37] Ellen M Voorhees et al. 1999. The TREC-8 question answering track report. In *Trec*.
[38] Adina Williams, Nikita Nangia, and Samuel R. Bowman. 2018. A Broad-Coverage Challenge Corpus for Sentence Understanding through Inference. In *NAACL*.
[39] Dustin Wright and Isabelle Augenstein. 2020. Claim check-worthiness detection as positive unlabelled learning. *arXiv preprint arXiv:2003.02736* (2020).
[40] Shuiqiao Yang, Jiaojiao Jiang, Arindam Pal, Kun Yu, Fang Chen, and Shui Yu. 2020. Analysis and Insights for Myths Circulating on Twitter During the COVID-19 Pandemic. *IEEE Open Journal of the Computer Society* 1 (2020), 209–219.
[41] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. XLNet: Generalized Autoregressive Pretraining for Language Understanding. In *NeurIPS*.
[42] Wenpeng Yin, Jamaal Hay, and Dan Roth. 2019. Benchmarking Zero-shot Text Classification: Datasets, Evaluation and Entailment Approach. In *EMNLP*.