Generating Fact Checking Briefs

Angela Fan[1,3], **Aleksandra Piktus**[1], **Fabio Petroni**[1], **Guillaume Wenzek**[1], **Marzieh Saeidi**[1], **Andreas Vlachos**[4], **Antoine Bordes**[1], **Sebastian Riedel** [1,2]

[1]Facebook AI Research [2]University College London [3]LORIA, [4]University of Cambridge
{angelafan,piktus,fabiopetroni,guw,marzieh,avlachos,abordes,sriedel}@fb.com

Abstract

Fact checking at scale is difficult—while the number of active fact checking websites is growing, it remains too small for the needs of the contemporary media ecosystem. However, despite good intentions, contributions from volunteers are often error-prone, and thus in practice restricted to claim detection. We investigate how to increase the accuracy and efficiency of fact checking by providing information about the claim before performing the check, in the form of natural language *briefs*. We investigate passage-based briefs, containing a relevant passage from Wikipedia, entity-centric ones consisting of Wikipedia pages of mentioned entities, and Question-Answering Briefs, with questions decomposing the claim, and their answers. To produce QABriefs, we develop QABRIEFER, a model that generates a set of questions conditioned on the claim, searches the web for evidence, and generates answers. To train its components, we introduce QABRIEFDATASET which we collected via crowdsourcing. We show that fact checking with briefs — in particular QABriefs — increases the accuracy of crowdworkers by 10% while slightly decreasing the time taken. For volunteer (unpaid) fact checkers, QABriefs slightly increase accuracy and reduce the time required by around 20%.

1 Introduction

Fact checking is a challenging task. It requires deep knowledge of a claim's topic and domain, as well as an understanding of the intricacies of misinformation itself. Checking a single claim can take professional fact checkers 15 minutes to one day (Hassan et al., 2015). Volunteers on the other hand are not considered accurate enough; with access to a search engine, Roitero et al. (2020) report crowdsourced fact check accuracies of around 58%. This result corroborates earlier reports[1] by fact checking websites which attempted to engage volunteers, but reported success only for claim detection, which is considered a much simpler task (Konstantinovskiy et al., 2018). This is problematic, both from the perspective of using crowdsourced fact checking to combat misinformation and from the perspective of helping individuals fact check themselves.

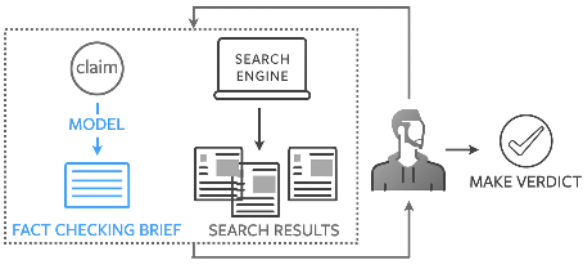

Figure 1: **Fact Checking Briefs.** Before conducting a fact check, we propose generating *briefs* to provide information about the claim. We show they make fact checking more accurate and efficient.

One path for scaling fact checking could be through full automation, taking a claim as input and producing a verdict (Vlachos and Riedel, 2014). Existing work has framed fact checking as classification, often supported by evidence (Wang, 2017; Thorne et al., 2018; Augenstein et al., 2019). However, due to the limitations of existing automated solutions, practitioners prefer solutions that improve efficiency in reaching a verdict, instead of approaches to the complete process (Graves, 2018).

In this work, we propose *briefs* to increase the accuracy and efficiency of fact checking (Figure 1). By generating fact checking briefs, our models aim to provide evidence a human fact checker would find useful. We investigate several approaches, in-

[1]http://mediashift.org/2010/11/crowdsourced-fact-checking-what-we-learned-from-truthsquad320/, http://fullfact.org/blog/2018/may/crowdsourced-factchecking/

Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, pages 7147–7161,
November 16–20, 2020.

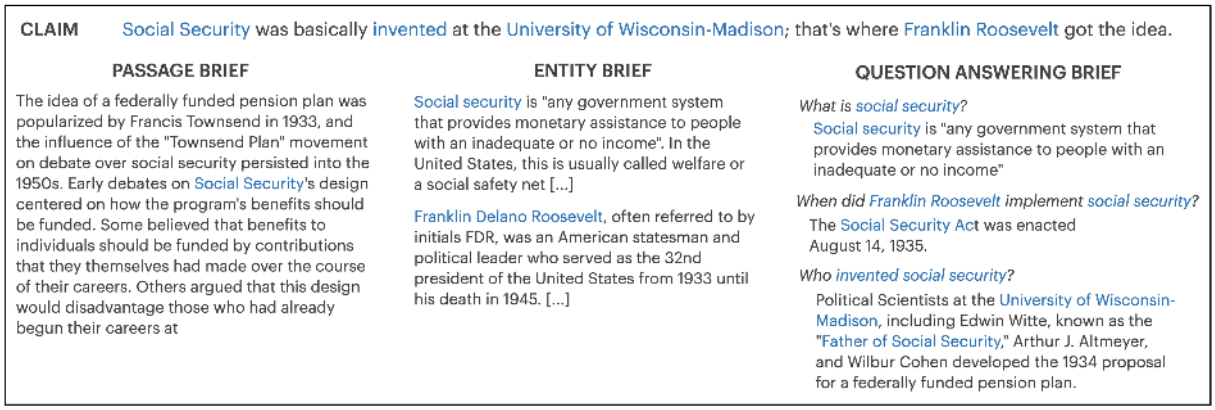

Figure 2: **Three Types of Briefs: (1) Passage Briefs**, based on information retrieval applied to the claim, **(2) Entity Briefs**, using entity linking to identify information about each entity, and **(3) Question Answering Briefs**, which condition on the claim to generate questions, then answer questions using open domain question answering

cluding returning Wikipedia passages that relate to the claim, and an entity linking approach that shows information about mentioned entities. Crucially, we introduce QABriefs — a set of relevant questions and their answers (see Figure 2).

To learn how to produce QABriefs and create training data, we use crowdsourcing to gather such briefs based on existing fact checks. We create QABRIEFDATASET, a collection of about 10,000 QABriefs with roughly 3 question and answer pairs each. We introduce QABRIEFER, a novel model that performs structured generation via claim-conditioned question generation and open domain question answering. Each question is used to identify evidence using a search engine. Finally, a pretrained question answering model is finetuned to generate answers and produce the full brief.

In experiments with crowdworkers, QABriefs improve accuracy by 10% compared to using only a search bar while reducing the time a fact check takes. For volunteer fact checkers, accuracy is improved by 4% and the process is 20% faster compared to using a search bar. Using QABriefs from human annotators leads to the largest improvement, followed by briefs generated by QABRIEFER and other proposed forms of briefs. This suggests that briefs are a promising avenue for improving crowdsourced fact checking. Further, QABRIEFDATASET can be used to develop models capable of answering challenging, real world questions.

2 Briefs for Fact Checking

Fact checkers must comprehend each part of a claim, which requires gathering information about a wide range of concepts— a precise definition of a term, how a politician voted, or the exact contents of a bill. Such knowledge is available in many sources: knowledge bases, statistical reports, or on the internet. We introduce the notion of *briefs* to provide relevant information to fact checkers—as if *briefing* them before fact checking— and explore three possible forms: Passage Briefs, Entity Briefs, and Question Answering Briefs. We show how they can be constructed with modern NLP approaches.

2.1 Passage Briefs

To provide information before checking a claim, Passage Briefs consist of relevant passages retrieved from Wikipedia. For the claim in Figure 2, information about the history and implementation of social security in the United States is retrieved and presented as background for the fact checker. To generate Passage Briefs, we identify relevant Wikipedia passages for each claim. Based on the results by Lewis et al. (2020) on open-Wikipedia tasks, we use the Dense Passage Retriever (DPR) (Karpukhin et al., 2020). This state of the art, pretrained retriever model learns a representation of questions and possible relevant paragraphs. In our case, we provide the claim as input instead of the question, rank the outputs, and select the top ranked passage. We limit to 500 tokens for readability. Initial experiments suggested web-based Passage Briefs returned poor results for most claims, as it relied on a finding a single passage addressing the entire claim, so we keep the Passage Brief focused on Wikipedia. Further, DPR is trained on Wikipedia, and we found the best performance within this domain.

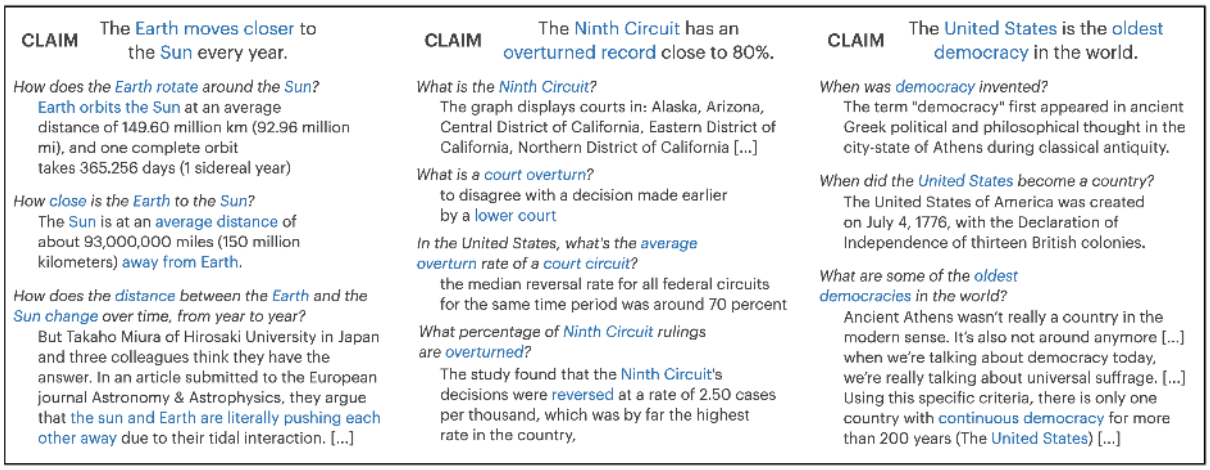

Figure 3: **Examples of QABriefs in QABRIEFDATASET**

2.2 Entity Briefs

Passage briefs provide information from a single passage, but claims are complex and often require multiple pieces of information from different sources. Thus we propose entity briefs that focus on each entity referenced in the claim.

Entities in each claim are identified with BLINK (Wu et al., 2019), a model trained on Wikipedia data that links each entity to its nearest Wikipedia page. BLINK combines a bi-encoder (Urbanek et al., 2019; Humeau et al., 2019) that identifies candidates with a cross-encoder that models the interaction between mention context and entity descriptions. For each entity, we retrieve its Wikipedia and provide the first paragraph in the brief. In Figure 2, *Franklin Roosevelt* is an entity, and the brief communicates he is *an American politician who served as the 32nd president of the United States [...]*. However, unlike Passage Briefs, if several entities are identified, information from multiple pages is displayed in an Entity Brief.

2.3 Question Answering Briefs

Entity briefs provide information about entities mentioned in the claim, but not necessarily the evidence needed for the claim in question. For this reason we propose QABriefs, which decompose fact checking into a set of questions and answers. E.g. the claim in Figure 2 could be split into understanding what social security is, identifying who invented the concept, and finally where Franklin Roosevelt got the idea. Each step can be written into a question — *What is social security? Who invented social security?* — that is then answered. The decomposition into question-answer pairs is likely to be better amenable to the current generation of information retrieval systems, which typically assume simpler information needs, e.g. most QA datasets have questions about single factoids. Unfortunately, there are no existing datasets or models available to create QABriefs. Next, we describe how we create a dataset (Section 3) and a model (Section 4) to produce QABriefs.

3 QABrief Dataset

To train and evaluate models to generate QABriefs, we collect a dataset of questions based on claims, together with answers to those questions found on the open web. Crucially, annotators first read the article from a fact checking website that describes how the claim was checked, and then decompose the process into questions, for which answers are provided. The claims for the dataset are sourced from existing fact checking datasets, specifically DATACOMMONS[2] and MULTIFC (Augenstein et al., 2019). The annotator instructions are in the Appendix and examples are shown in Figure 3.

While numerous question generation and answering datasets exist, none of them focuses on using questions and answers to combat misinformation. QABRIEFDATASET focuses on this real world problem, with each question grounded in a claim that was actually fact checked. Further, existing datasets are quite different from our usecase — for example, many datasets are based on Wikipedia, but fact checkers find evidence from other sources. Many datasets have short answer spans, but our questions are complex, so require longer answers.

[2]https://datacommons.org/factcheck

3.1 Question Generation

Crowdworkers are asked to read a *claim* and its corresponding *fact checking article*[3], which details the investigative process used to perform the fact check. After reading the article, crowdworkers write *questions* to reconstruct the process taken by professional fact checkers. For each claim, crowdworkers write two to five questions that are at least five words long and standalone. For instance, the question *why did he do that* is invalid, as it is not clear what *he* or *that* is. We discourage questions with yes/no answers and discourage questions about the same claim from overlapping more than five words.

After the questions are collected, a *question validation* phase is conducted. A separate group of crowdworkers reviews the quality of the questions and flags those that are redundant and/or otherwise poor quality. For example, questions such as *What evidence is there that [claim] is true?* are rejected. Other instances of questions rejected at this phase include nonsensical questions and questions that simply rephrase the claim. Any questions that do not pass this review are re-annotated. Finally, a *question clarity* phase is conducted — crowdworkers read the questions and edit those that are unclear or underspecified. For example, questions may need to have a year added to them to accurately verify a statistic. Further, additional questions can be added if crowdworkers feel the existing questions are not sufficient. This can lead to more than five questions per claim. Spelling errors are highlighted and crowdworkers are encouraged to correct them.

3.2 Question Answering

After each claim is annotated with multiple questions, we proceed to collect the answers to them. To answer questions, crowdworkers are given the claim; the source of the claim (for example, the entity who said the quote being checked); and the question. Crowdworkers enter a *query* into a *search engine* to find information on the web. The search is restricted from accessing fact checking domains, to prevent the answer from being trivially found on a fact checker's website. The query does not need to be identical to the question, and is often rephrased to find better search results. After reading the returned results, crowdworkers can provide one of three possible answer types:

[3]For our running example, the reference article is: https://www.politifact.com/factchecks/2016/dec/16/russ-feingold/was-social-security-basically-invented-university-/

Train	Number of Claims	5,897
	Number of QA Pairs	18,281
Valid	Number of Claims	500
	Number of QA Pairs	1,431
Test	Number of Claims	500
	Number of QA Pairs	1,456
Avg Number Questions/Claim		3.16
Avg Number Words in Questions		10.54
Avg Number Words in Answers		43.56

Table 1: **Statistics of QABRIEFDATASET**

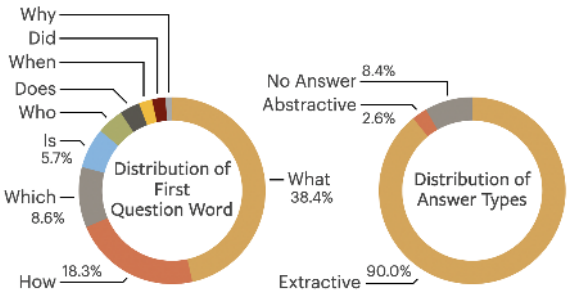

Figure 4: **Question and Answer Types**

- *Extractive* — the encouraged option, crowdworkers copy paste up to 250 words as an answer. We focus on extractive answers, as identifying such an answer is more straightforward compared to writing an answer.
- *Abstractive* — if the answer is present in an image or graph, crowdworkers write an abstractive answer of at least 20 words.
- *No Answer* — if no answer can be found, crowdworkers write an explanation of at least 20 words to describe why there is no answer.

Next, *validation* is conducted. The questions are complex, so we do not assume the answer is known. Crowdworkers instead flag answers that seem incorrect. For example, if the answer to *How many people live in California* is *three billion*, this would be flagged and re-annotated. A last step is conducted for answers that are *No Answer*. To verify that answers cannot be found, a second group of crowdworkers tries to find an answer. If an answer is found, the *No Answer* annotation is discarded.

3.3 QABrief Dataset Statistics

In summary, QABRIEFDATASET includes 6,897 claims and 21,168 questions paired with their answers. We use 500 claims as a validation set and 500 claims as a test set. The validation and test sets include around 1400 questions and answers each.

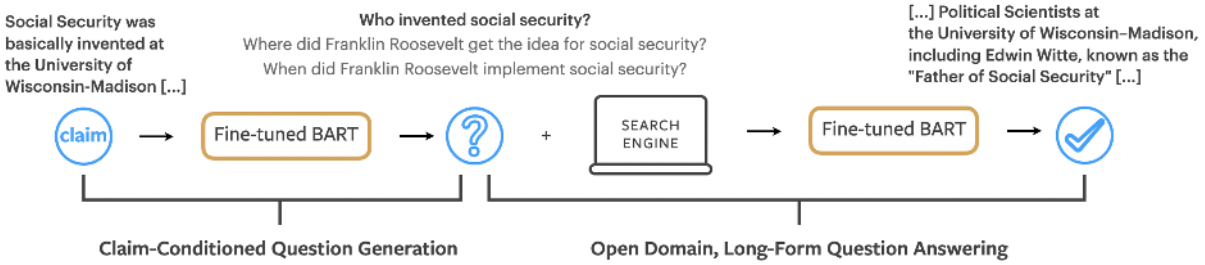

Figure 5: **QABriefer Model.** First, BART is finetuned to conduct claim-conditioned question generation and generates a sequence of questions that decompose a fact check. Second, we use an information retrieval system and a second finetuned BART model to extract long-form answers to each question.

We examine the types of questions to analyze the diversity. Table 1 shows that each claim on average requires around 3 questions to cover the different parts of the claim, and questions contain 10.5 words on average. The questions are quite diverse, as seen in Figure 4 (left), though the majority begin with *What, How, Which* question words. There are few *Why* questions, indicating a focus on verifying factual information, rather than causality.

The answers obtained have mainly extractive annotations, though a small portion of abstractive and no answer options exist (see Figure 4, right). Answers are around 43.5 words long (Table 1), though abstractive answers are generally shorter as crowdworkers must fully write them.

We examined a subset of 50 claims where we conducted multiple data collection trials with the same claim to understand the agreement rate between workers. We found that for the question annotation step, about half of the questions provided by different people on the same claim were very similar and could be considered paraphrases. For example, the questions *Who invented social security* and *Who was the invetor of social security*. For the answer annotation step, the identified answers varied in length but were often paraphrases — some crowdworkers tended to select only the specific span that answered the question (e.g. an entity name), while others chose several sentences to capture the context.

4 QABrief Model

The automatic generation of QABriefs presents numerous modeling challenges. Generating such a brief is a hierarchical process: writing the questions, and then conditioned upon the questions, searching the web and writing the answers. While many question answering datasets exist, questions in QABRIEFDATASET are grounded on real claims that were fact checked. The diversity of the claims renders reusing questions across claims unlikely to work, thus precluding the use of retrieve-and-rank approaches (Rao and Daumé III, 2018). Unlike previous question generation models (Du et al., 2017; Duan et al., 2017; Tang et al., 2017; Zhao et al., 2018) that generate based on an answer, we treat question generation closer to structured planning — laying out the format for the entire brief.

In contrast to most question answering datasets, the length of the answers in QABRIEFDATASET are long-form (Fan et al., 2019). For example, the average answer in SQuAD (Rajpurkar et al., 2016) is four words long, while the average answer in QABRIEFDATASET is forty. Further, datasets such as SQuAD, Natural Questions (Kwiatkowski et al., 2019), and HotpotQA (Yang et al., 2018) are built from Wikipedia, while QABriefs uses the web.

In this section, we describe QABRIEFER (see Figure 5). For each claim, the question generation model is used to generate multiple questions. For each question, an evidence document is retrieved using a search engine. We take the top search hit as the evidence and retrieve the text from CommonCrawl[4]. Finally, the generated question and retrieved evidence document is provided to the question answering model to generate an answer.

4.1 Question Generation

The first step of QABRIEFER is to create the questions that will form the structure of the brief. To create models that can take a claim as input and generate a sequence of questions as output, we use sequence-to-sequence (Sutskever et al., 2014) models. As QABRIEFDATASET is not large enough to train the language model needed for question generation, we leverage advances in pretraining and use QABRIEFDATASET to adapt it to the task at

[4] http://commoncrawl.org/

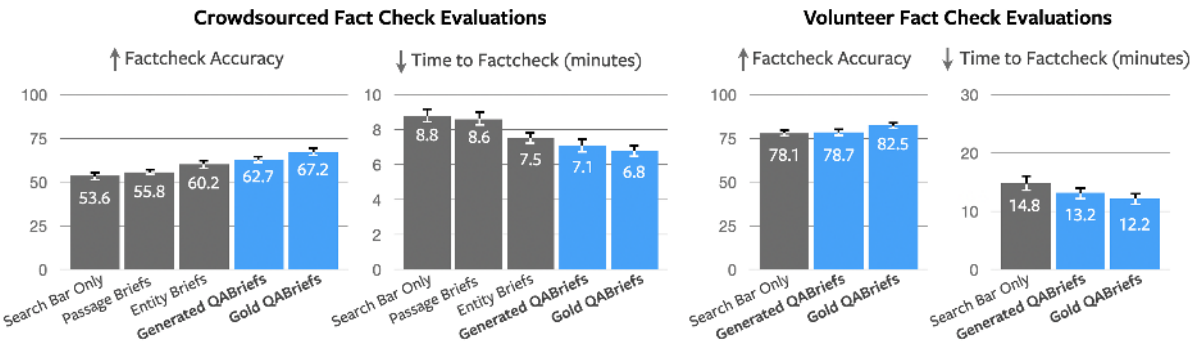

Figure 6: **Accuracy and Time Taken to fact check** by Crowdworkers (left) and Volunteer fact checkers (right). Briefs of various forms, but particularly QABriefs, increase fact checking accuracy and efficiency.

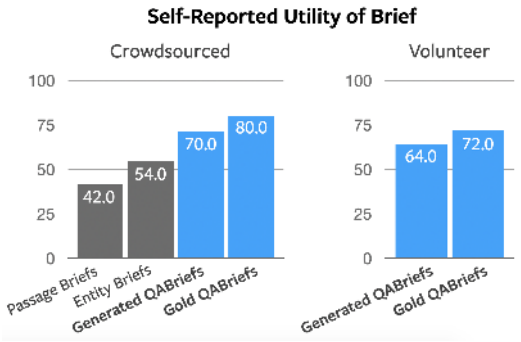

Figure 7: **Usefulness of Briefs** reported by Crowdsourced and Volunteer Fact Checkers.

hand. We use BART (Lewis et al., 2019), a denoising autoencoder that uses various noise functions and trains to recreate the input. In adapting BART for question generation based on claims, we explore three options: generating all questions based only on the claim, generating all questions based on the claim and the source of the claim (usually an entity), and generating questions one at a time. To write questions one at a time, the model conditions on the previous questions as well as the claim and source, and needs to predict the subsequent question or an *end of questions* token.

4.2 Question Answering

Given the question-based structure for QABriefs, the second part of the hierarchical process is to identify answers. Models take as input the question and evidence document that annotators indicated to contain the answer, and produce an answer. As QABRIEFDATASET does not have enough data to train a question answering model from scratch, we use BART finetuned on Natural Questions. and subsequently finetune it further on QABRIEFDATASET. As the dataset contains extractive and abstractive answers as well as questions where the model must provide an explanation to justify no answer, we use an abstractive approach with a generative model; abstractive models have shown strong performance on various question answering tasks (Lewis and Fan, 2018; Dong et al., 2019; Radford et al.; Raffel et al., 2019; Lewis et al., 2020).

5 Experimental Setup

Our main question is whether briefs can increase the accuracy and efficiency of fact checking. We focus on human evaluation with both crowdworkers and volunteers fact checking claims.

5.1 Human Evaluation

Metrics We evaluate the *accuracy* of a fact check by comparing the verdict from our human evaluators with professionals. The professional fact checking labels are obtained from the DATACOMMONS dataset. We measure the *time* taken to fact check from when the task is loaded to when the verdict and explanation is submitted.

Crowdsourced Evaluators Crowdworkers on Mechanical Turk are presented with the 500 test set claims and instructed to use a search bar to decide if the claim is true, false, or in the middle. They then write at least 20 words justifying their verdict. We indicate that if a claim is *mostly true* it should be labeled as true, and *mostly false* should be false. We discourage the middle option and suggest it should be used only if a verdict cannot be made, to prevent it from being the default. Previous work has shown that fine-grained labels, such as *sometimes true*, *half true*, *mostly true* are difficult to calibrate even with professional fact checkers (Lim, 2018), so we opt for a more simpler scale. The search bar queries the open web, but is restricted from searching known fact checking domains. Evaluators either use only the search bar, or are provided

Model	BLEU
Claim ⇒ Qs	12.8
Claim + Source ⇒ Qs	13.2
Claim + Source + Prev Qs ⇒ Next Q	13.4

Table 2: **Question Generation Models**

Model	F1
BART FT on QABRIEFDATASET	30.5
BART FT on NQ + QABRIEFDATASET	32.8

Table 3: **Question Answering Models.**

with a brief to read before the fact check. The same claims are evaluated with all methods. We repeat the study three times to assess variance.

Volunteer Evaluators Crowdsourced evaluation is scalable, but crowdworkers may be less motivated to spend a large amount of time fact checking. Thus, we conduct a smaller scale study using graduate student volunteer evaluators, recruited by asking for those interested in the challenge of fact checking real claims themselves. Volunteers are presented with 100 claims rather than 500, but otherwise conduct the same task as crowdworkers. Volunteers compare the search-bar-only fact checking process with generated QABriefs and gold QABriefs. We do not evaluate Passage Briefs or Entity Briefs, as we found volunteer fact checking to be less scalable than crowdsourcing.

5.2 Automatic Evaluation of Model Quality

To evaluate the quality of question generation, following existing work (Duan et al., 2017), we use BLEU. To evaluate the quality of question answering, we use F1 score (Rajpurkar et al., 2016).

5.3 Model Details

We use `fairseq-py` (Ott et al., 2019) to train the QABRIEFER. We use the open-sourced BART model (Lewis et al., 2019) and suggested finetuning hyperparameters, training for 10 epochs and taking the best epoch by validation loss. To generate, we use beam search with beam size 5. We tune the length penalty to decode such that written questions and answers approximately match the average length in the validation split. Exact training and generation commands, with further experimental details, can be found in the appendix.

6 Results

We show in human evaluations that fact checking efficiency and accuracy are improved with briefs.

6.1 Briefs Increase Fact Checking Quality

We examine the accuracy of crowdsourced and volunteer fact checkers when presented —in addition to a search bar— with different types of briefs: Passage, Entity, and QABriefs. For QABriefs, we examine briefs generated by QABRIEFER and the *Gold* briefs annotated in QABRIEFDATASET. We compare briefs against a *search bar only* baseline.

As shown in Figure 6 (left), when crowdworkers are presented with briefs, fact checking accuracy increases, even when taking into account variance in three repeated trials. The Passage Briefs are not more helpful in terms of accuracy compared to using the search bar alone, but Entity Briefs and QABriefs are both better than this baseline. Providing Gold rather than generated QABriefs performs best — suggesting modeling improvements could help bridge the gap. For crowdworkers, using briefs slightly reduces the time taken (from 8.8 minutes on average to around 7), but the overall time spent is low compared to professionals, who spend from 15 minutes to one day (Hassan et al., 2015).

For volunteer fact checkers (Figure 6, right), accuracy across all methods is higher compared to crowdworkers. Providing the Gold QABrief remains the best, though the gap is smaller than for crowdworkers. Providing the QABrief slightly decreases time taken to fact check. Note that the average volunteer spends twice the amount of time compared to a crowdworker, and this thoroughness probably contributes to higher accuracy, as well as the smaller improvement from providing briefs.

6.2 QABriefs are Preferred

Next, we further contrast QABriefs with Passage and Entity Briefs. We ask evaluators to consider if the brief made the fact check *easier* or provided useful *background context*. Crowdworkers rated QABriefs helpful twice as often as Passage Briefs (In Figure 7). When evaluators submit a fact check, they must write an explanation for their reasoning. Qualitatively examining these, we found many references to the QABrief. Evaluators noted that *based on [the QABrief], I searched for [X evidence]*. We hypothesize that the question-answer format may be easier to read, as it is naturally organized and possibly less redundant.

6.3 Generating QABriefs with QABRIEFER

Lastly, we assess the performance of our proposed QABRIEFER model. We display the BLEU scores for our proposed Question Generation models in Table 2 and find that iteratively writing questions one by one is the best performing method. Further, providing information about the source of the claim (usually the entity who made the claim) provides better results. Question Answering results are shown in Table 3. We find that first fine-tuning on a large question answering dataset, Natural Questions (NQ), and further fine-tuning on QABRIEFDATASET provides the best results. Likely, this is because BART is a general purpose generative model, so fine-tuning for question answering first on a much larger dataset is useful.

7 Related Work

Previous work in NLP has focused on claim veracity. It has been treated as a classification problem (Wang, 2017), often using stance detection (Riedel et al., 2017). The FEVER Challenge (Thorne et al., 2018) proposed providing provenance for a decision along with classification, and various approaches developed combine information retrieval with stance detection or question answering (Li et al., 2018; Lee et al., 2018). Question generation and answering has been considered in the context of FEVER (Jobanputra, 2019) — the focus was on eliciting the right answer from a question answering system rather than improving the accuracy and efficiency of human fact checkers.

However, FEVER is based on modified Wikipedia sentences, not real world claims, which are arguably more difficult. To address this Hanselowski et al. (2019) considered the claims fact checked by the website Snopes, but used the reports accompanying them as evidence instead of finding the evidence directly. Popat et al. (2018) and Augenstein et al. (2019) used search engines, but without ensuring that they provide evidence supporting/refuting the claim instead of being related to it or that they were not fact checking reports. Finally, Kochkina et al. (2018) used responses on social media for rumour verification, but did not address evidence finding.

Various work studies how to improve the fact checking process. Analysis shows accuracy can improve by providing feedback (Hill, 2017), additional time (Bago et al., 2020), tooling (Karduni et al., 2019), or training (Zhang et al., 2018).

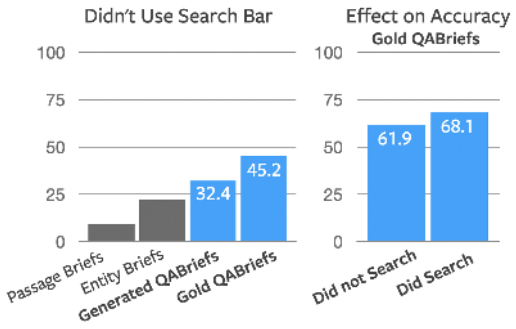

Figure 8: **Overconfidence** when given a QABrief.

These works are complementary to ours — we provide support in the form of briefs. Studies emphasize that current solutions for fully automated fact checking face various challenges (Graves, 2018) that must be addressed with interdisciplinary research (Karduni, 2019). Developing tools to aid human-in-the-loop fact checking has received increasing attention, from NLP to human-computer interaction and psychology, often with positive results when tested with journalists (Miranda et al., 2019) and professionals (Lurie, 2019).

8 Discussion

While our experiments show a generally positive impact of briefs for human fact checking, it is important to put them into a broader perspective.

Briefs for Professional Fact Checkers Crowdworkers and professional fact checkers perform different tasks under very different circumstances. Professionals often investigate alternative interpretations and produce an explanation of their process in an article. They often have years of experience and must check a variety of claims. Consequently, we do not claim that briefs will make a difference in their work. Nevertheless, QABriefs can provide insights into the fact checking process. As the QABrief dataset was created using professional fact checking articles describing how a claim was checked, by decomposing a claim into multiple components, we can encourage a more structured fact checking process.

Biases introduced by Briefs While briefs can increase accuracy, they can introduce biases. We found that providing a QABrief increased confidence — many submitted their fact check based on the brief alone, without the search bar. Figure 8 (left) displays that around 45% of crowdworkers did not use the search bar when given the Gold

QABrief, even though accuracy without the search bar is reduced. Briefs aid accuracy and efficiency, but are not fully sufficient to produce a verdict.

Metrics for Factchecking We focus on improving fact checking accuracy, but we note that agreement amongst professionals is not 100% (Lim, 2018). Professionals often agree if part of a claim is true or false, but disagree on the importance (Lim, 2018) or pursue different directions for checking the claim (Marietta et al., 2015; Amazeen, 2016). Different fact checkers have different scales, which are not calibrated. Nevertheless, improving the accuracy of crowd sourced fact checkers is still reflective of agreement with professionals.

9 Conclusion

We propose the concept of fact checking briefs, to be read before performing a fact check. Crucially, we develop QABRIEFER and release the accompanying QABRIEFDATASET, to create QABriefs. We show in extensive empirical studies with crowdworkers and volunteers that QABriefs can improve accuracy and efficiency of fact checking.

References

Michelle A Amazeen. 2016. Checking the fact-checkers in 2008: Predicting political ad scrutiny and assessing consistency. *Journal of Political Marketing*, 15(4):433–464.

Isabelle Augenstein, Christina Lioma, Dongsheng Wang, Lucas Chaves Lima, Casper Hansen, Christian Hansen, and Jakob Grue Simonsen. 2019. MultiFC: A real-world multi-domain dataset for evidence-based fact checking of claims. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4685–4697, Hong Kong, China. Association for Computational Linguistics.

Bence Bago, David G Rand, and Gordon Pennycook. 2020. Fake news, fast and slow: Deliberation reduces belief in false (but not true) news headlines. *Journal of experimental psychology: general*.

Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019. Unified language model pre-training for natural language understanding and generation. In *Advances in Neural Information Processing Systems*, pages 13042–13054.

Xinya Du, Junru Shao, and Claire Cardie. 2017. Learning to ask: Neural question generation for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1342–1352.

Nan Duan, Duyu Tang, Peng Chen, and Ming Zhou. 2017. Question generation for question answering. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 866–874.

Angela Fan, Yacine Jernite, Ethan Perez, David Grangier, Jason Weston, and Michael Auli. 2019. Eli5: Long form question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3558–3567.

D Graves. 2018. Understanding the promise and limits of automated fact-checking.

Andreas Hanselowski, Christian Stab, Claudia Schulz, Zile Li, and Iryna Gurevych. 2019. A richly annotated corpus for different tasks in automated fact-checking. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 493–503, Hong Kong, China. Association for Computational Linguistics.

Naeemul Hassan, Bill Adair, James Hamilton, Chengkai Li, Mark Tremayne, Jun Yang, and Cong Yu. 2015. The quest to automate fact-checking. *Proceedings of the 2015 Computation + Journalism Symposium*.

Seth J Hill. 2017. Learning together slowly: Bayesian learning about political facts. *The Journal of Politics*, 79(4):1403–1418.

Samuel Humeau, Kurt Shuster, Marie-Anne Lachaux, and Jason Weston. 2019. Poly-encoders: Transformer architectures and pre-training strategies for fast and accurate multi-sentence scoring. *CoRR abs/1905.01969. External Links: Link Cited by*, 2:2–2.

Mayank Jobanputra. 2019. Unsupervised question answering for fact-checking. In *Proceedings of the Second Workshop on Fact Extraction and VERification (FEVER)*, pages 52–56, Hong Kong, China. Association for Computational Linguistics.

Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. *arXiv preprint arXiv:1705.03551*.

Alireza Karduni. 2019. Human-misinformation interaction: Understanding the interdisciplinary approach needed to computationally combat false information. *arXiv preprint arXiv:1903.07136*.

Alireza Karduni, Isaac Cho, Ryan Wesslen, Sashank Santhanam, Svitlana Volkova, Dustin L Arendt, Samira Shaikh, and Wenwen Dou. 2019. Vulnerable to misinformation? verifi! In *Proceedings of the 24th International Conference on Intelligent User Interfaces*, pages 312–323.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. *CoRR*, abs/2004.04906.

Elena Kochkina, Maria Liakata, and Arkaitz Zubiaga. 2018. All-in-one: Multi-task learning for rumour verification. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3402–3413, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Lev Konstantinovskiy, Oliver Price, Mevan Babakar, and Arkaitz Zubiaga. 2018. Towards automated factchecking: Developing an annotation schema and benchmark for consistent automated claim detection.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466.

Nayeon Lee, Chien-Sheng Wu, and Pascale Fung. 2018. Improving large-scale fact-checking using decomposable attention models and lexical tagging. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1133–1138, Brussels, Belgium. Association for Computational Linguistics.

Mike Lewis and Angela Fan. 2018. Generative question answering: Learning to answer the whole question.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Kuttler, Mike Lewis, Wen-tau Yih, Tim Rocktaschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks.

Sizhen Li, Shuai Zhao, Bo Cheng, and Hao Yang. 2018. An end-to-end multi-task learning model for fact checking. In *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*, pages 138–144, Brussels, Belgium. Association for Computational Linguistics.

Chloe Lim. 2018. Checking how fact-checkers check. *Research & Politics*, 5(3):2053168018786848.

Emma Lurie. 2019. The challenges of algorithmically assigning fact-checks: A sociotechnical examination of google's reviewed claims.

Morgan Marietta, David C Barker, and Todd Bowser. 2015. Fact-checking polarized politics: Does the fact-check industry provide consistent guidance on disputed realities? In *The Forum*, volume 13, pages 577–596.

Sebastião Miranda, David Nogueira, Afonso Mendes, Andreas Vlachos, Andrew Secker, Rebecca Garrett, Jeff Mitchel, and Zita Marinho. 2019. Automated fact checking in the news room. In *The World Wide Web Conference*, pages 3579–3583.

Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. Ms marco: a human-generated machine reading comprehension dataset.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53.

Kashyap Popat, Subhabrata Mukherjee, Andrew Yates, and Gerhard Weikum. 2018. DeClarE: Debunking fake news and false claims using evidence-aware deep learning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 22–32, Brussels, Belgium. Association for Computational Linguistics.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392.

Sudha Rao and Hal Daumé III. 2018. Learning to ask good questions: Ranking clarification questions using neural expected value of perfect information. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2737–2746, Melbourne, Australia. Association for Computational Linguistics.

Benjamin Riedel, Isabelle Augenstein, Georgios P Spithourakis, and Sebastian Riedel. 2017. A simple but tough-to-beat baseline for the fake news challenge stance detection task. *arXiv preprint arXiv:1707.03264*.

Kevin Roitero, Michael Soprano, Shaoyang Fan, Damiano Spina, Stefano Mizzaro, and Gianluca Demartini. 2020. Can the crowd identify misinformation

objectively? the effects of judgment scale and assessor's background. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

Duyu Tang, Nan Duan, Tao Qin, Zhao Yan, and Ming Zhou. 2017. Question answering and question generation as dual tasks. *arXiv preprint arXiv:1706.02027*.

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

Jack Urbanek, Angela Fan, Siddharth Karamcheti, Saachi Jain, Samuel Humeau, Emily Dinan, Tim Rocktäschel, Douwe Kiela, Arthur Szlam, and Jason Weston. 2019. Learning to speak and act in a fantasy text adventure game. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 673–683.

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

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

Ledell Wu, Fabio Petroni, Martin Josifoski, Sebastian Riedel, and Luke Zettlemoyer. 2019. Zero-shot entity linking with dense entity retrieval. *arXiv preprint arXiv:1911.03814*.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380.

Amy X Zhang, Aditya Ranganathan, Sarah Emlen Metz, Scott Appling, Connie Moon Sehat, Norman Gilmore, Nick B Adams, Emmanuel Vincent, Jennifer Lee, Martin Robbins, et al. 2018. A structured response to misinformation: Defining and annotating credibility indicators in news articles. In *Companion Proceedings of the The Web Conference 2018*, pages 603–612.

Yao Zhao, Xiaochuan Ni, Yuanyuan Ding, and Qifa Ke. 2018. Paragraph-level neural question generation with maxout pointer and gated self-attention networks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3901–3910.

10 Appendix

10.1 Dataset Analysis

In this section, we describe qualitative observations about QABRIEFDATASET to provide more insight.

How are fact checks decomposed into questions? We analyze the strategies taken by annotators to decompose the fact checking process of a claim into multiple questions. There are several distinct strategies that emerge:

For questions about comparison, annotators usually write 1-2 questions validating the first part of the comparison and 1-2 questions validating the second part of the comparison.

For questions about historical events, annotators usually clarify the entities involved and clarify the background. Annotators often ask questions about time and location. Several questions of the form *Did X event really happen* arise, but are often filtered by later steps of the dataset collection process (see description later in this Appendix).

For questions about what an individual may have said, annotators adopt a strategy very similar to professional fact checkers. A common trend in misinformation is misattribution, or saying an individual said a statement when they did not. Often, a misalingment in time or location can reveal this — if the person was not yet born, for example. Annotators often ask many questions to try to uncover this.

How are annotators finding answers? In many standard question answering datasets, the question-answer pairs already exist. For example, in TriviaQA (Joshi et al., 2017), the questions and answers are from Trivia enthusiasts, and in ELI5 (Fan et al., 2019), the questions and answers are from Reddit question answering subreddits. Other datsets collect questions and answers, but focus on identifying extractive answers in Wikipedia, an arguably easier task than finding them on the web. In SQuAD (Rajpurkar et al., 2016), questions are often written by modifying a sentence of Wikipedia into a question. In Natural Questions (Kwiatkowski et al., 2019) and MSMarco (Nguyen et al., 2016), the questions are real questions submitted to Google and Bing search engines, but the answers are much more straightforward (short, extractive spans).

In contrast, QABRIEFDATASET faces challenges because the questions are complex and the answers must be found on the open web. In initial experiments, we attempted to restrict only to Wikipedia, but found that a large quantity of the questions were annotated with *No Answer*. To find answers on the web is a difficult task, as many answers depend heavily on context. Checking statistics, for example, is particularly difficult, as the year must be correct. We focus on using automated checks, described later on in this Appendix, to check for high quality answers. Further, we spot checked answers manually for quality control.

We analyzed the main strategies taken to find answers. About 50% of the annotators directly enter the question in the search bar, but the other 50% mainly use keyword searches to find better results. Around 83% of annotators only use the search bar once, but the rest use the search bar two to four times to refine their search query. Note this search query data will be released as part of QABRIEFDATASET as well.

Most annotators submit an answer from the first three search results. Unfortunately, our interface cannot capture how many search results they opened and read before submitting a response. If Wikipedia was in the top search result, most annotators tended to submit a response from Wikipedia.

10.2 Additional Human Evaluation Results

In this section, we present additional results from our human evaluation studies. We contrast the process taken by professionals with our volunteer evaluators, analyze if evaluators can accurately assess how difficult a claim is to fact check, and display more detailed results to examine the time taken to fact check a claim.

Fact Checking Process of Non-Professionals In contrast to professionals, we find that crowdworkers and volunteer fact checkers often act on more general understanding rather than validating every detail. For example, for some claims, explanations written for a verdict included *It's not possible because the government cannot enforce*, but no evidence is cited. Over-reliance on common sense can lead to less evidence-based decision-making, and most likely contributes to less time-intensive checks compared to professionals. Another instance that commonly arises is checking certain statistics, such as *how many people purchased X item*. A professional fact checker will cross-reference the year carefully, examine how purchases are quantified in stores and through online retailers, and break it down by country. A volunteer examining the same claim will investigate with a search engine, but likely trust a holistic

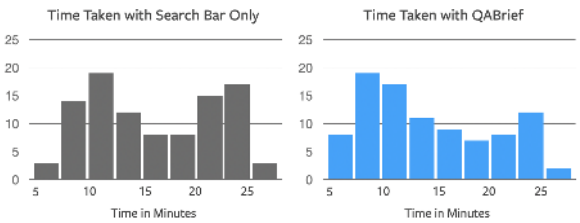

Figure 9: **Time Taken** to fact check by volunteer evaluators. The distribution is bimodal.

number they find, rather than breaking it down.

Self-Reported Fact Check Difficulty We found that crowdworkers and volunteer fact checkers were not accurate at assessing the difficulty of a fact check, and their assessments of difficulty did not correspond well to accuracy. We ask each fact checker to report the perceived difficulty of the process, either *easy, medium, hard* before they submitted their verdict. We found that their self-reported perceived difficulty did not correlate with their accuracy — even if evaluators felt the claim was *easy*, they were only 4% more accurate in accurately checking it. For *medium* and *hard* claims, the accuracy of fact checking was the same.

Time Taken to Factcheck We present additional results for volunteer fact checking and the time taken to examine claims. As shown in Figure 9, the time taken to fact check is bimodal, most likely because certain claims are easier and others require detailed investigation. Easier claims that were submitted more quickly tended to be checks based more on common sense, for example to fact check the claim *Shark found swimming on freeway in Houston*. When given QABriefs, the distribution of time taken shifts to smaller quantities.

10.3 Model Training Details

In this section, we provide detailed information about the training procedure of QABRIEFER as well as exact training and generation parameters used in `fairseq-py` to produce our results.

Question Generation We use the open sourced BART-large model. We finetune with learning rate $3e-05$, maximum tokens 2048 per batch, warming up for 500 updates and training for 10 epochs. Models are trained with label smoothing 0.1 and dropout 0.1. For optimization, we use the Adam optimizer and train with weight decay 0.01. We tuned only the dropout parameter, between values $0.1, 0.2, 0.3$, but otherwise took these parameters from the suggested parameter settings for BART finetuning. After training, we choose the best checkpoint by validation loss. The total training time is 8 hours on 1 GPU, though reasonable performance is reached after about 5 hours of training. As our model is finetuned BART large, it retains the same parameter count of 406M parameters.

For generation, we generate with beam size 5. We tune the length penalty between $0.5, 1, 1.5, 2$ and adjust the minimum and maximum length parameters. For minimum length, we examined values between $3, 5, 10$ and for maximum length, we examined values between $20, 30, 40, 50, 60$. To select the best generation hyperparameters, we generated on the validation set and chose the hyperparameters that maximized BLEU on validation to use on the test set.

Question Answering We use the open sourced BART-large model. We finetune with learning rate $3e-05$, maximum tokens 2048 per batch, warming up for 500 updates and training for 10 epochs. Models are trained with label smoothing 0.1 and dropout 0.1. For optimization, we use the Adam optimizer and train with weight decay 0.01. We use the suggested parameter settings for BART finetuning. After training, we choose the best checkpoint by validation loss. The total training time is 8 hours on 1 GPU, though reasonable performance is reached after about 7 hours of training. As our model is finetuned BART large, it retains the same parameter count of 406M parameters.

For generation, we generate with beam size 5, tuning the beam size between $4, 5$. We keep the length penalty fixed to 1. We adjust the minimum length parameter between $10, 50$. We adjust the maximum length parameter between $50, 100, 250$. To select the best generation hyperparameters, we generated on the validation set and chose the hyperparameters that maximized BLEU on validation to use on the test set.

10.4 Dataset Collection Details

In this section, we provide additional details on the instructions given to crowdworkers when constructing QABRIEFDATASET and describe all steps. Figure 10 illustrates the full dataset collection process.

10.4.1 Recruitment for the Task

We used the crowdworking platform Amazon Mechanical Turk. Evaluators were provided with the task and instructions, and could look at the task and

Figure 10: **QABRIEFDATASET**: Annotators read the claim and professionally written fact checking article describing how the claim was checked. Questions and answers are annotated and validated for quality control. Questions are edited so each question is standalone, and *No Answer* options are verified.

opt to decline. For volunteer fact checkers, volunteers were given a description of our goals and the task they would perform. Volunteers were asked if they were interested in fact checking.

10.4.2 Question Generation Data Collection

Instructions for Writing Questions: Our goal is to understand how a claim is fact checked. Perform the following steps:

- *Read* the claim and the article that describes the fact checking process from a professional fact checker.
- *Think* what questions the fact checker had to answer to reach a verdict for the claim
- *Write* 3-5 questions that reflect the fact checking process used to reach a verdict.
- Questions must be *standalone* — do not write questions that refer to other questions, specify the names of people/places, etc

Important!

- *DO NOT* write questions with yes or no answers
- *DO NOT* write questions that rephrase the claim

Must Read Examples:

- *Good:* What was the population of California in 2000?
- *Bad:* What was the population of California? [No time specified to find a statistic]

- *Good:* How many education bills did Senator Smith vote for in March, 2000?
- *Bad:* How many education bills did he vote for? [Who is he? Also no time specified]

- *Good:* How do sharks move around?
- *Bad:* Is it true that sharks can walk on land? [Yes or no question, and directly asks if something is true or not]

In this data collection step, we used a number of automatic checks implemented into the task. Annotators could not submit without filling out at least 3 questions, each of at least 5 tokens in length. The questions could not overlap with each other more than 5 words. The questions could not exactly match the claim. Annotators could not submit in the first minute of the task. For each problem detected by the automatic check, an error message was displayed explaining why the current submission was not valid.

Instructions for Validating Questions : Our goal is to understand the steps necessary to fact check a claim. Perform the following steps:

- *Read* the claim and the article that describes the fact checking process
- *Read* the questions that describe the steps taken by the fact checker to reach a verdict
- *Write* additional questions **or** *Choose* no additional questions needed

Additional question writing guidelines are the same as for the original writing questions step. Annotators that write more questions are paid a bonus.

Instructions for Question Clarity : Our goal is to make sure each question is readable and could be used in a Google search to find an answer. Perform the following steps:

- *Read* the question
- Do you think the question could be Googled to find an answer? If not, *read* the article and add more detail to the question

Must Read Examples:

- *Original:* What was the population of California?
- *Edit:* What was the population of California in 2000? [Adds year]

- *Original:* How many education bills did he vote for?
- *Edit:* How many education bills did Senator Smith vote for in March, 2000? [Adds name and year]

10.4.3 Question Answering Data Collection

Instructions for Finding Answers: Our goal is to find answers to each of these questions. Perform the following steps:

- *Read* the question
- Use the *Search Bar* to find an answer
- If you cannot find an answer, you must write an explanation why you cannot find the answer

Important!

- *DO NOT* use any other search bar to find an answer. You MUST use the provided search bar only.
- *Do NOT* answer the claim or predict a verdict. Your job is to find an answer to the *QUESTION*
- *DO NOT* submit answers from politifact.com or factcheck.org. These answers will not be accepted. If you use our provided search bar, you will not have this problem. Use the provided search bar!

The task then has a dynamic workflow, which we now describe. After using the search bar, annotators had to select between one of three options:

- I found an answer, and I can copy paste the text of the answer from the webpage
- I cannot copy paste the answer because it is in a graph, table, or picture, but I can write the answer myself.
- I cannot find an answer. I understand I will need to write an explanation why an answer cannot be found

and these options correspond to extractive, abstractive, and no answer possibilities.

If the annotator chose the first option, an extractive answer, they were presented with a form with the following instructions: copy-paste the answer text, copy-paste the URL the answer is from. They are asked to *Copy paste the answer. DO NOT copy paste the entire site, only the part that answers the question. You can paste a maximum of 250 words*.

If the annotator chose the second option, an abstractive answer, they were presented with a form with the following instructions: write the answer text using at least 20 words, copy-paste the URL the answer is from.

If the annotator chose the third option, no answer, they were presented with a form with the following instructions: write an explanation for why no answer can be found using at least 20 words.

In this data collection step, we used a number of automatic checks implemented into the task. Annotators could not submit the task unless all requested areas (based on their chosen branch of the workflow) were filled out. The extractive answer could not be more than 250 words in length and could not be the empty string (one word answers were accepted). The abstractive answer and no answer explanation had to be at least 20 words in length. The copy pasted URL the annotators submitted as evidence for their answer had to match the URLs of their returned search results. This serves a dual purpose check — first that annotators used our search bar, which is restricted from accessing fact checking domains, and second that annotators submitted a real URL. Annotators could not submit in the first minute of the task. Annotators could not submit URLs that were known fact checking domains. For each problem detected by the automatic check, an error message was displayed explaining why the current submission was not valid.