The Dissertation Committee for Jifan Chen
certifies that this is the approved version of the following dissertation:

Building Robust and Modular Question Answering Systems

Committee:

Gregory Durrett, Supervisor

Eunsol Choi, Co-Supervisor

Raymond J. Mooney

Daniel Andor

Building Robust and Modular Question Answering Systems

by

Jifan Chen

DISSERTATION

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT AUSTIN

May 2023

Acknowledgments

Time flies so quickly; it feels like just yesterday when I arrived in Austin, and now I am sitting at my "old" familiar desk, completing this dissertation.

I would like to start by expressing my deepest gratitude to my principal supervisor, Greg Durrett, for his unwavering guidance, support, and mentorship throughout this journey. I first met Greg in Berlin at ACL 2016 when he presented a poster at the conference. He explained his complex summarization system to me; although I didn't fully grasp it, I found it quite intriguing. At that time, I had no idea that he would become my Ph.D. advisor. As one of his first students, I have been consistently impressed by his passion and enthusiasm for research. He encouraged me to pursue research topics that people would find interesting after several years, regardless of whether they are mainstream or not. In fact, he believed that distancing ourselves from the mainstream would help our work stand out. Greg was always patient, sharp, and clear when discussing research ideas, and I have learned so much from him (more than I realized). One valuable lesson I learned from him is you have to gain a deep understanding of the task you are working on before you can propose something truly meaningful, and you need some tries and errors to get this understanding.

I would also like to extend my heartfelt thanks to my co-supervisor, Eunsol Choi, for her valuable insights, constructive feedback, and continuous encouragement. The first time I got to know Eunsol was her job talk at UT. At that time, I was truly impressed by her presentation and the way she conveyed her research ideas. Our first collaboration was in 2021 when we were working on the answer

verification project. She was always careful and was able to find the blind spots in the project. Since then, she became my official co-advisor. I was often shocked by how many projects she can simultaneously handle and her fast context-switching abilities. I am grateful to have had the opportunity to learn from her expertise. Not only has she been an incredible source of knowledge and inspiration, but she has also shown genuine care and concern for my well-being and career development.

I would also like to thank Ray Mooney and Daniel Andor for serving on my dissertation committee. Ray is a very "old school" professor and I truly appreciate his comments on my dissertation and talk, which have helped me take a deeper dive into my work. Daniel was my mentor during my internship at Google, and I am grateful that he agreed to be one of the committee members. Whenever I talk with him, he raises razor-sharp questions that point out where the problems lie.

I am grateful to my lab mates at TAUR Lab. Yasu: we joined the lab together, and I can't quite recall how many research and casual talks we've had. My Ph.D. life would have been much more boring without you. Jiacheng: we knew each other since our time at Fudan and reunited here in TAUR Lab. You're such a smart person, and I always enjoy our fun and relaxed chats. Pengxiang: you have been my mentor in navigating Austin life. Rouhan: I genuinely enjoyed our weekly basketball games. Kaj: working with you on the final course project as TAs was an absolute pleasure. Juan Diego and Manya: we met during my last year of Ph.D. study and I truly appreciated your support during the time I was preparing for my final defense. Shih-ting Lin: we finished the reasoning chain for the multi-hop question answering paper together and it was good to work with you.

I would like to acknowledge my friends. Jiacheng Zhuo, Jiacheng Xu, and Xingyi Zhou: I will always cherish the good old "2221" times. Su Wang: you are one of my best friends in Austin. You are like a big brother to me, and I always

feel at ease when we spend time together, as if I were at home. Yuhao Zhang: you are both my mentor and friend during my Amazon internship. I've learned a lot from you, both in research and in the 8-ball pool. I got a better understanding of "the devil is in the details" from you. Yian Zhang: we met during the internship at Amazon, and I enjoyed the chat with you on multiple topics. Kaiser Sun: I would always remember the "fish catching" time we shared during the Amazon internship.

My deepest appreciation goes to my parents, Mei and Hao, for their unconditional love, support, and belief in my abilities. They are ordinary teachers in a small town in southwest China. They don't know what exactly I am doing but they always felt proud of what I've achieved. They always encouraged me to pursue what I want and be the person I want to be. I haven't been able to go home for more than three years since COVID started and I wanted to reunite with them soon. My sincere appreciation goes to Mengning, my girlfriend. We met during a swimming class in my second year, and since then, we have shared four amazing years together. Our shared love for travel has taken us to numerous destinations, including Paris, Spain, Cancun, Alaska, Hawaii, Maine, Colorado, and many other places. My Ph.D. journey would not have been as vibrant without her, and I hope we can continue to explore the world together in the future.

Finally, I would like to thank all those who have directly or indirectly contributed to my research and have not been mentioned here. Your assistance, support, and encouragement have not gone unnoticed, and I am grateful for your contributions to the successful completion of this dissertation.

Building Robust and Modular Question Answering Systems

Publication No. _____

Jifan Chen
The University of Texas at Austin, 2023

Supervisors: Gregory Durrett and Eunsol Choi

Over the past few years, significant progress has been made in QA systems due to the availability of annotated datasets on a large scale and the impressive advancements in large-scale pre-trained language models. Despite these successes, the black-box nature of end-to-end trained QA systems makes them hard to interpret and control. When these systems encounter inputs that deviate from their training data distribution or are subjected to adversarial perturbations, their performance tends to deteriorate by a large margin. Furthermore, they may occasionally produce unanticipated results, potentially leading to confusion among users. Additionally, this deficiency in robustness and interpretability poses challenges when deploying such models in real-world scenarios.

In this dissertation, we aim to build robust QA systems by explicitly decomposing various QA tasks into distinct sub-modules, each responsible for a particular aspect of the overall QA process. Through this decomposition, we seek to achieve improved performance in terms of both the system's ability to handle diverse and challenging inputs (robustness) and its capacity to provide transparent and explainable reasoning (interpretability).

To address the aforementioned limitations, in this dissertation, we aim to build robust QA models by explicitly decomposing different QA tasks into different sub-modules. We argue that utilizing these sub-modules can substantially improve the robustness and interpretability of different QA systems. In the first half of this dissertation, we introduce three sub-modules to mitigate the dataset artifacts that models learn from datasets. These sub-modules also enable us to examine and exert explicit control over the intermediate outputs. In the first work, to address question answering that requires multi-hop reasoning, we propose a chain extractor, which extracts the reasoning chains necessary for models to derive the final answer. The reasoning chains not only prevent the model from exploiting reasoning shortcuts but also provide an explanation of how the answer is derived. In the second work, we incorporate an alignment layer between the question and the context before generating the answer. This alignment layer can help us interpret the models' behavior and improve the robustness of adversarial settings. In the third work, we add an answer verifier after QA models generate the answer. This verifier can boost QA models' prediction confidence across several different domains and help us spot cases where QA models predict the right answer for the wrong reason by utilizing the external NLI datasets and models.

In the second half of this dissertation, we tackle the problem of complex fact-checking in the real world by treating it as a modularized QA task. We first decompose a complex claim into several yes-no subquestions whose answer directly contributes to the veracity of the claim. Then, each sub-question is fed into a commercial search engine to retrieve relevant documents. Additionally, we extract the relevant snippets in the retrieved documents and use a GPT3-based summarizer to generate the core evidence for checking the claim. We show that the decompositions can play an important role in both evidence retrieval and veracity composition of

an explainable fact-checking system. Also, we show the GPT3-based evidence summarizer generates faithful summaries of documents most of the time indicating it can be used as an effective part of the pipeline. Moreover, we annotate a dataset – ClaimDecomp, containing 1,200 complex claims and the decompositions. We believe that this dataset can further promote building explainable fact-checking systems and analyzing complex claims in the real world.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Question answering (QA) is one of the core tasks of natural language processing (NLP) due to its wide-ranging applications, including but not limited to virtual assistants, customer support, and educational platforms. In recent years, advancements in deep learning techniques (Seo et al., 2017; Peters et al., 2018; Devlin et al., 2019; Liu et al., 2019; Raffel et al., 2020a; Brown et al., 2020; Wei et al., 2021; Ouyang et al., 2022) and the availability of large-scale annotated datasets (Rajpurkar et al., 2016b; Yang et al., 2018; Joshi et al., 2017a; Kwiatkowski et al., 2019; Reddy et al., 2019; Clark et al., 2020) have led to huge progress in QA systems. For example, Chung et al. (2022) showed that a 5-shot Flan-PaLM model outperforms the average human rater on 57 MMLU (Hendrycks et al., 2020) tasks while also approaching the performance of the average human expert.

Despite the great successes, even the SOTA QA models like GPT3 (Brown et al., 2020) are still not generally robust – when testing on challenging settings like adversarially perturbed datasets (Jia and Liang, 2017; Wallace et al., 2019; Gardner et al., 2020; Bartolo et al., 2020), they sometimes fail to generate the right answer and their performance tends to decrease by a notable margin (Liang et al., 2022). Also, they sometimes produce unreliable outputs that may mislead users. Figure 1.1 and Figure 1.2 show two examples of unreliable outputs from Google and New Bing Search (powered by GPT-3.5).

Over the past few years, prior work has tackled the problem in various

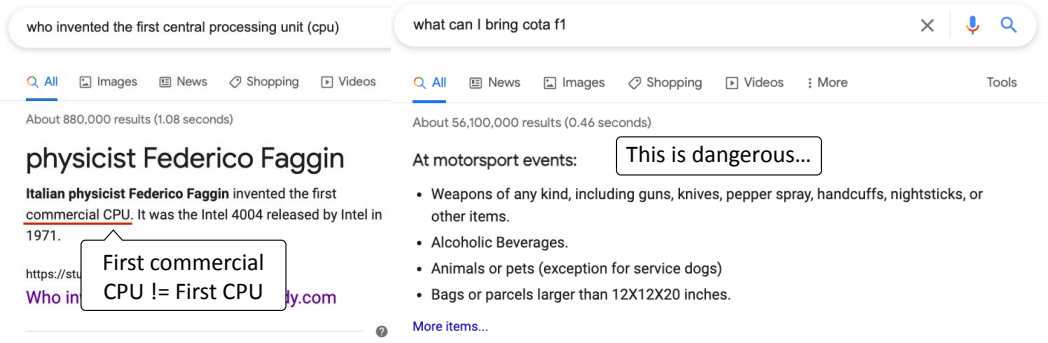


Figure 1.1: Example of cases where the answers returned by the QA system is not reliable. In the first example, the model ignores "commercial"; in the second example, the content returned may even cause trouble.

ways, including data augmentation and adversarial training (Wang and Bansal, 2018; Khashabi et al., 2020; Liu et al., 2020), generative QA (Lewis and Fan, 2018), debiasing the training data (Utama et al., 2020; Swayamdipta et al., 2020; Gardner et al., 2021), and adding regularizers (Yeh and Chen, 2019; Zhou et al., 2020). We take a different angle: modularizing different QA tasks and constructing a pipelined QA system by assembling sub-modules. There are two main advantages by using sub-modules. First, the sub-modules can be learned through external resources such as external models, data, and knowledge. Second, by checking the intermediate output from the sub-modules, we can better understand and gauge the QA models' behavior, e.g., why a particular prediction is made and how can we potentially avoid bad predictions. The proposed framework is illustrated in Figure 1.3.

In Chapter 3, we propose a reasoning chain extractor as a sub-module to tackle questions which require multi-hop reasoning. The reasoning chain extractor can extract a discrete reasoning chain over the text, which consists of a series of sentences leading to the answer. Critically, we do not rely on gold annotated chains or "supporting facts": at training time, we derive pseudogold reasoning chains using

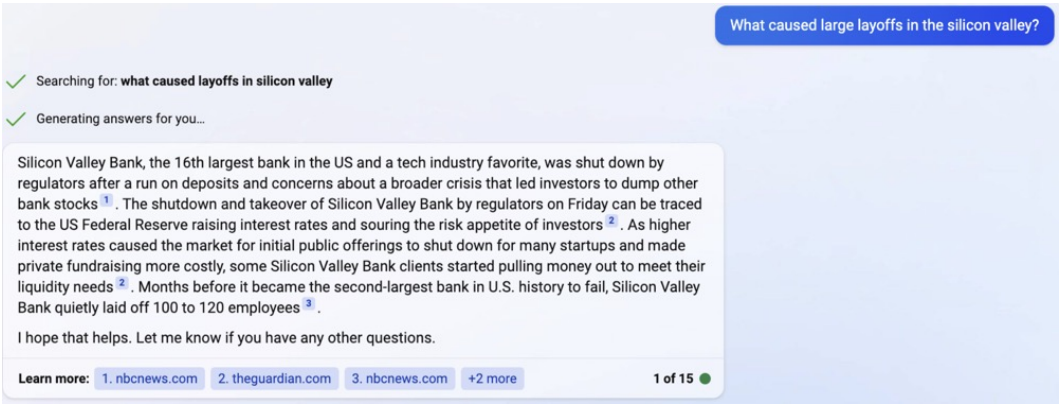


Figure 1.2: Example of output returned by the New Bing Search where the model simply misunderstands the question and returns an answer about "Silicon Valley Bank" instead of Silicon Valley. This example is taken on 03/23/2023.

heuristics based on named entity recognition and coreference resolution. Nor do we rely on these annotations at test time, as our model learns to extract chains from raw text alone. Our analysis shows properties of chains that are crucial for high performance: in particular, modeling extraction sequentially is important, as is dealing with each candidate sentence in a context-aware way. Furthermore, human evaluation shows that our extracted chains allow humans to give answers with high confidence, indicating that these are a strong intermediate abstraction for this task.

In Chapter 4, we model question answering as an alignment problem. We decompose both the question and context into smaller units based on off-the-shelf semantic representations (here, semantic roles), and align the question to a subgraph of the context in order to find the answer. We formulate our model as a structured SVM, with alignment scores computed via BERT, and we can train end-to-end despite using beam search for approximate inference. Our use of explicit alignments allows us to explore a set of constraints with which we can prohibit certain types of bad model behavior arising in cross-domain settings. Furthermore, by investigating differences in scores across different potential answers, we can seek to understand

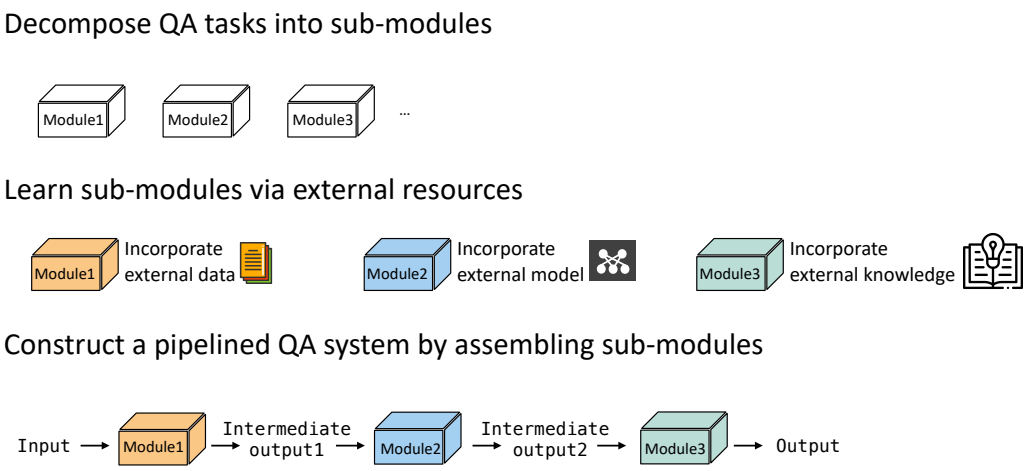


Figure 1.3: Illustration of the proposed framework where we first decompose a QA task into sub-modules; each sub-module is learned through external resources and then assembled to construct a pipelined QA system.

what particular aspects of the input lead the model to choose the answer without relying on post-hoc explanation techniques.

In Chapter 5, we explore the use of natural language inference (NLI) as a verifier to check QA models' predictions. We leverage large pre-trained models and recent prior datasets to construct powerful question conversion and decontextualization modules, which can reformulate QA instances as premise-hypothesis pairs with very high reliability. Then, by combining standard NLI datasets with NLI examples automatically derived from QA training data, we can train NLI models to evaluate QA systems' proposed answers. We show that our approach improves the confidence estimation of a QA model across different domains. Careful manual analysis over the predictions of our NLI model shows that it can further identify cases where the QA model produces the right answer for the wrong reason.

In Chapter 6, we tackle the problem of complex fact-checking in the real-world by treating it as a modularized QA task. Specifically, we present ClaimDe-

COMP, a dataset of decompositions for 1,200 claims. Given a claim and its verification paragraph written by fact-checkers, our trained annotators write subquestions covering both explicit propositions of the original claim and its implicit facets, such as additional political context that changes our view of the claim's veracity. We study whether state-of-the-art pretrained models can learn to generate such subquestions. Our experiments show that these models generate reasonable questions, but predicting implied subquestions based only on the claim (without consulting other evidence) remains challenging. Nevertheless, we show that predicted subquestions can help identify relevant evidence to fact-check the full claim and derive the veracity through their answers, suggesting that claim decomposition can be a useful piece of a fact-checking pipeline

In Chapter 7, we extend Chapter 6 by retrieving the raw evidence using our decomposed sub-questions in the wild and building the full pipeline for real-world fact-checking. To simulate the realistic fact-checking scenario, we conduct our experiments where the retriever can only search documents available prior to the statement of the claim, following the real-world use case that fact-checkers would face. Our whole pipeline includes five components: claim decomposition, raw document retrieval, fine-grained evidence retrieval, evidence aggregation (using GPT-3), and veracity judgment. We conduct experiments on CLAIMDECOMP, the dataset we created in Chapter 6, and show that the aggregated evidence produced by our pipeline not only improves the performance of veracity judgment over the no-evidence baseline but also can be used as evidence to help human fact-checkers to make their decisions. Finally, we show that the veracity classification performance of our system is bottlenecked by web retrieval, and building a human-machine-in-the-loop fact-checking system is a promising future direction.

Chapter 2

Background and Related Work

This chapter presents the related work and background supporting this dissertation. We first introduce a line of work that explore the brittleness of the recent QA systems, then we discuss previous literature that focuses on improving the robustness of QA systems for different aspects. Additionaly, we discuss modular QA systems, which decompose the end-to-end trained QA models into several sub-modules. These sub-modules often provides better robustness and interpretability.

2.1 Brittleness in QA systems

As machine learning became the dominant strategy for building QA systems in recent decades, the behavior of such systems heavily relies on training data (Jia, 2020). When the systems are tested on out-of-distribution data, they are often unreliable and generate undesired outputs. A growing body of research has focused on exploring the brittleness in QA systems under various conditions.

Perturbation-based attacks One natural way to test models' robustness is to slightly perturb samples from the training distribution. These perturbations include character-level alterations, such as typos (Belinkov and Bisk, 2018; Ebrahimi et al., 2018a; Piktus et al., 2019). Typos present a significant practical challenge, as they frequently occur in natural text but typically do not hinder human comprehension. Other perturbations involve word substitution with synonyms or neighboring words

in vector space (Alzantot et al., 2018; Iyyer et al., 2018; Jia et al., 2019; Jin et al., 2020). For instance, Ribeiro et al. (2018) showed that by simply replacing "?" with "??" in examples from the SQuAD development set led to 202 more errors by a state-of-the-art model at that time, this alone increased the overall error rate by 3%. Further, entity-level perturbations have been explored (Balasubramanian et al., 2020; Yan et al., 2022). For instance, Yan et al. (2022) showed that current pre-trained language models sometimes solve entity-related questions based on the learned knowledge rather than the context. Replacing the entities with unseen ones causes a drop in the model performance. Also, there are researches exploring sentence-level perturbation (Jia and Liang, 2017; Wallace et al., 2019; Gardner et al., 2020). For example, Wallace et al. (2019) showed that by injecting specific trigger strings, the model can always output offensive content regardless of the question.

Recently, large-pre-trained language models such as GPT-3 (Brown et al., 2020; Ouyang et al., 2022) have brought great progress in language understanding and established SOTA performance on a wide range of NLP benchmarks. However, they are still not robust against those perturbations as demonstrated by Liang et al. (2022).[1]

Natural distribution shifts Another line of work focuses on addressing how natural distribution shifts affect QA models' performance. Unlike perturbation-based attacks which sometimes generate unreal or deliberately adversarial/hard examples, natural distribution shifts better represent the real-world scenario when deploying QA models. There are two common types of natural distribution shifts: domain generalization and subpopulation shift (Koh et al., 2021). For domain generaliza-

[1]See the HELM leaderboard for the performance of various models in robustness setting: `https://crfm.stanford.edu/helm/latest/?group=core_scenarios`

tion, the test distribution comes from related but distinct domains of the training distribution. For subpopulation shifts, the test distributions are subpopulations of the training distribution.

Fisch et al. (2019) adapted and unified 18 distinct question-answering datasets into the same format to benchmark the cross-domain generalization ability of QA models. Among them, six datasets were made available for training, six datasets were made available for development, and the rest were hidden for final evaluation. Kiela et al. (2021); Ma et al. (2021); Zhang and Choi (2021); Thrush et al. (2022) proposed to collect test samples in a dynamic way, which can be viewed as a stress test for models' robustness under subpopulation shift. Koh et al. (2021); Sagawa et al. (2022) proposed WILDS and its extension, a curated benchmark of 10 datasets reflecting a diverse range of distribution shifts that naturally arise in real-world applications.

2.2 Addressing Brittleness in QA systems

Data Augmentation Data augmentation is a simple but effective way to achieve stronger generalization. Wang and Bansal (2018); Liu et al. (2020); Khashabi et al. (2020); Gardner et al. (2020) demonstrate that by adding automatically perturbed examples during training, models are more robust against perturbation-based attacks and domain shifts while keeping the same performance for in-domain data. Furthermore, Dua et al. (2021); Bartolo et al. (2021); Lee et al. (2020) showed that generating synthetic adversarial question-answer pairs not only improved the model robustness against adversarial attack but also improved model generalization across various domains.

Modular QA systems Another line of research to build robust QA systems focuses on decomposing the model into sub-modules so that the whole model is less likely to be biased by the training data distribution. (Hu et al., 2019; Kamath et al., 2020; Wang et al., 2020b; Zhang et al., 2021) introduced an extra answer verification layeras a final step for question answering models. The answer verifiers substantially improve the cross-domain generalization of models under a selective question-answering setting where models only choose to answer top-k percent of questions it most confident with. Talmor and Berant (2018); Min et al. (2019b); Perez et al. (2020) showed that by decomposing the complex questions into simple ones, QA models are more robust against reasoning shortcuts in multi-hop reasoning. (Wolfson et al., 2020) further introduces a Question Decomposition Meaning Representation (QDMR) to explicitly model this process. The work we presented in this dissertation generally falls into this vein where we add sub-modules to different QA tasks to mitigate the biases model learned and enforce explicit control over models' behavior.

Chapter 3

Multihop Reasoning via Reasonning Chains

This chapter is based on Chen et al. (2019a).[1]

3.1 Introduction

As high performance has been achieved in simple question answering settings (Rajpurkar et al., 2016b), work in this area has increasingly gravitated towards questions that require more complex reasoning to solve. Multi-hop question answering datasets explicitly require aggregating clues from different parts of some given documents (Dua et al., 2019; Welbl et al., 2018; Yang et al., 2018; Jansen et al., 2018; Khashabi et al., 2018a). Earlier question answering datasets contain some questions of this form (Richardson et al., 2013; Lai et al., 2017), but typically exhibit a limited range of multi-hop phenomena. Designers of multi-hop datasets aim to test a range of reasoning types (Yang et al., 2018) and, ideally, systems should have to behave in a very specific way in order to do well. However, Chen and Durrett (2019) and Min et al. (2019a) show that models achieving high performance may not actually be performing the expected kinds of reasoning. Partially this is due to the difficulty of evaluating intermediate model components such as attention (Jain and Wallace, 2019), but it also suggests that models may need inductive bias if they are to solve

[1] Jifan Chen, Shih-ting Lin, Greg Durrett. Multi-hop Question Answering via Reasoning Chains. 2019. arXiv preprint arXiv:1910.02610.
Jifan Chen initialized the research project, conducted experiments, analyzed data and wrote the paper.

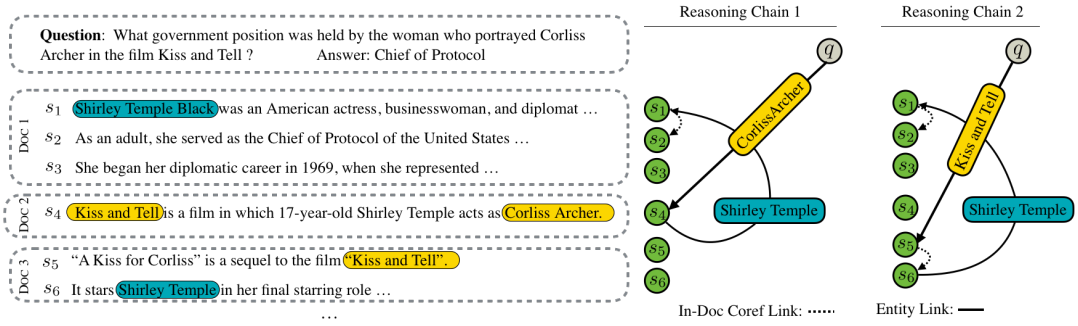


Figure 3.1: A multi-hop example chosen from the HotpotQA development set. Several documents are given as context to answer a question. We show two possible "reasoning chains" that leverage connections (shared entities or coreference relations) between sentences to arrive at the answer. The first chain is most appropriate, while the second requires a less well-supported inferential leap.

this problem "correctly."

In this chapter, we propose a step in this direction, with a two-stage model that identifies intermediate *reasoning chains* and then separately determines the answer. A reasoning chain is a sequence of sentences that logically connect the question to a fact relevant (or partially relevant) to giving a reasonably supported answer. Figure 3.1 shows an example of what such chains look like. Extracting chains gives us a discrete intermediate output of the reasoning process, which can help us gauge our model's behavior beyond just final task accuracy. Formally, our extractor model scores sequences of sentences and produces an n-best list of chains via beam search.

To find the right answer, we need to maintain uncertainty over this chain set, since the correct one may not immediately be evident, and for certain types of questions, information across multiple chains may even be relevant. Sifting through the retrieved information to actually identify the answer requires deeper, more expensive computation. We employ a second-stage answer module, a BERT-based QA

system (Devlin et al., 2019), which can be run cheaply given the pruned context. Our approach resembles past models for coarse-to-fine question answering (Choi et al., 2017; Min et al., 2018; Wang et al., 2019), but explores the context in a sequential fashion and is trained to produce more principled reasoning chains.

To train our model, we heuristically label examples with reasoning chains. We use a search procedure leveraging coreference and named entity recognition (NER) to find a path from the start sentence to an end sentence through a graph of related sentences. Constructing this graph requires running an NER system at train time, but does not rely on the answer or answer candidates (Kundu et al., 2018). Our system also does not require these annotations at test time, operating instead from raw text.

Our chain identification is effective and flexible: we can use it to derive supervision on two existing datasets. On HotpotQA (Yang et al., 2018), we found that these derived chains are essentially as effective as the ground-truth supporting fact provided by the dataset. In terms of final question answering accuracy, on the WikiHop dataset (Welbl et al., 2018), our approach achieves state-of-the-art performance by a substantial margin among the published systems, and on HotpotQA, we achieve strong results and outperform several recent published systems.

Our contributions are as follows: (1) We present a method for extracting oracle reasoning chains for multi-hop reasoning tasks. These chains generalize across multiple datasets and are comparable to human-annotated chains. (2) We present a model that learns from these chains at train time and at test time can produce a list of chains. Those chains could be used to gauge the behaviors of our model. (3) Results on two large datasets show strong performance of our chain extraction and show that the extracted chains are intrinsically a good representation of evidence for question answering.

3.2 Question Answering via Chain Extraction

We describe our notion of chain extraction in more detail. A reasoning chain is a sequence of sentences that logically connect the question to a fact relevant to determining the answer. Two adjacent sentences in a reasoning chain should be intuitively related: they should exhibit a shared entity or event, temporal structure, or some other kind of textual relation that would allow a human reader to connect the information they contain.

Figure 3.1 shows an example of possible reasoning chains of an real example. In this case, we need to find information about the actor who played *Corliss Archer* in *Kiss and Tell*. These question entities may appear in multiple places in the text, and it is generally difficult to know which entity mentions might eventually lead to text containing the answer. If we arrive at s_4 and find the new entity *Shirley Temple*, we then need to determine what government position she held, which in this case can be found by two additional steps. Other reasoning chains could theoretically lead to this answer, such as the second chain: Shirley Temple starred in the sequel to Kiss and Tell, which might lead us to infer that Shirley Temple also plays Corliss Archer in Kiss and Tell. Although less justified, we also view this as a valid reasoning chain. However, in general, there are also "connected" sequences of sentences that don't imply the answer; for example, they are connected by an entity which is not related to the question.

In determining this chain, we largely used information about entity coreference to connect the relevant pieces: either cross-document coreference about Shirley Temple or resolution of various pronouns. Another relevant cue is that subsequent information about Shirley Temple in Document 1 occurs later in the discourse, which in this case reflects temporal structure. However, solving coreference or temporal relation extraction in general is neither necessary nor sufficient to do chain extrac-

tion. Therefore, we design our system so that it does not rely on coreference at test time, but can instead directly extract reasoning chains based on what it has learned at training time.

Having established this notion of a reasoning chain, we have three questions to answer. First, how can we automatically select pseudo-ground-truth reasoning chains? Second, how do we model the chain extraction process? Third, how do we take one or more extracted chains and turn them into a final answer? We answer these three questions in the next section.

3.3 Learning to Extract Chains

3.3.1 Heuristic oracle chain construction

Following the intuition in Figure 3.1, we assume that there are two relevant connections between sentences that can form reasoning chains. First, the presence of a shared entity often implies some kind of connection. This is not always a sufficient clue, since common entities like *United States* may occur in otherwise unrelated sentences; however, because this oracle is only used at train time, it does not need to be 100% reliable for the model to learn a chain extraction procedure. Second, we assume that any two sentences in the same paragraph are connected; this is often true on the basis of coreference or other kinds of bridging anaphora.

We derive heuristic reasoning chains by searching over a graph which is constructed based on these factors. Each sentence s_i is represented as a node i in the graph. We run an off-the-shelf named entity recognition system to extract all entities for each sentence. If sentence i and sentence j contain a shared entity, we add an edge between node i and j. We then also add an edge between all pairs of

sentence within the same paragraph.[2]

Starting from the question node, we do an exhaustive search to find all possible chains that could lead to the answer. This process yields a set of possible chains with different lengths; two examples are shown in Figure 3.1. We use two different criteria to select heuristic oracles:

- **Shortest Path**: We simply take the shortest chain from the chain set as our oracle.
- **Question Overlap**: We compute the `ROUGE-1 F1` score for each chain's sentences with respect to the question and take the chain with the highest score. This encourages selection of more complete answer chains which address all of the question's parts without finding shortcuts.

3.3.2 Chain extraction model

Our chain extractor takes the input documents and questions as input and returns a variable-length sequence of sentence pointers as output. The processing flow of our chain extractor can be divided into two main parts: sentence encoding and chain prediction as shown in Figure 3.2.

Sentence Encoding Given a document containing n paragraphs and a question, we first concatenate the question with each paragraph and then encode them using the pre-trained BERT encoder (Devlin et al., 2019). We denote the encoded ith paragraph as p_i. We also encode the question by itself with BERT, denoting as

[2]We do not explicitly run a coreference system here since current coreference systems often introduce spurious arcs. Moreover, cross-document links can nearly always be found by exact string match, and since we add all within-paragraph links, exactly determining the coreference status of every mention is not needed.

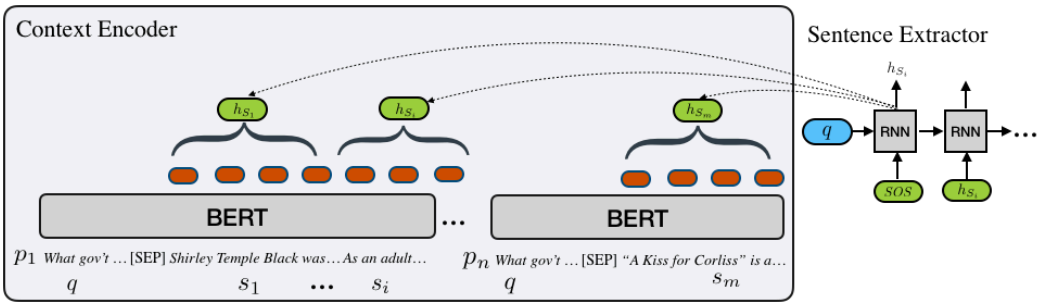


Figure 3.2: The BERT-Para variant of our proposed chain extractor. Left side: we encode each document paragraph jointly with the question and use pooling to form sentence representations. Right side: we use a pointer network extracts a sequence of sentences.

q. To compute the representation of a sentence, we extract it from the encoded paragraph. Suppose sentence k in the document is the jth sentence of paragraph i. Then $\mathbf{s}_k = \text{SpanExtractor}(p_i, s_j^{\text{START}}, s_j^{\text{END}})$. For simplicity, we choose max-pooling as our span extractor, though other choices are possible. We name this scheme of sentence representation as `BERT-Para`. This paragraph-factored model is much more efficient and scalable than attempting to run BERT on the full context, as full contexts can be thousands of words long. We also explore an even more factored version where each sentence is concatenated with the question and encoded independently, which we denote as `BERT-Sent`. Finally, instead of using BERT as the sentence encoder, we can use a bidirectional attention layer between the passage and question (Seo et al., 2017) as a baseline; we call this model `BiDAF-Para`.

Chain Prediction We treat all the encoded sentence representations as a bag of sentences and adopt an LSTM-based pointer network (Vinyals et al., 2015) to extract the reasoning chain, shown on the right side of Figure 3.2. At the first time step, we initialize the hidden state $\mathbf{h}_0$ in the pointer network using the max-pooled representation of the question q, and feed a special token `SOS` as the first input.

Let $c_1, \ldots, c_l$ denote the indices of sentences to include in the reasoning chain. At time step t, we compute the probability of sentence i being chosen as $P(c_t = i|c_1, \ldots, c_{t-1}, \mathbf{s}) = \text{softmax}(\alpha)[i]$, where $\alpha_i = \mathbf{W}[\mathbf{h}_{t-1}; \mathbf{s}_{c_{t-1}}; \mathbf{h}_{t-1} \odot \mathbf{s}_{c_{t-1}}]$, and $\mathbf{W}$ is a weight matrix to be learned.

Training the Chain Extractor During training, the loss for time step t is the negative log likelihood of the target sentence c_t^* for that time step: $\text{loss}_t = -\log(P(c_t^*|c_1^*, \ldots, c_{t-1}^*\mathbf{s}))$. We also explored training with reinforcement learning to optimize downstream prediction accuracy. For the two datasets we considered, pre-training with our oracle and fine-tuning with policy gradient did not lead to an improvement. Pure oracle chain extraction appears strong enough for the model to learn the needed associations across chain timesteps, but this may not be true on other datasets.

At evaluation time, we use beam search to explore a set of possible chains, which results in a set of chains $\mathbf{c}_1, \mathbf{c}_2, ..., \mathbf{c}_k$, with each chain containing different number of sentences.

3.3.3 Answer prediction

Since different beams may contain different plausible reasoning chains as shown in Figure 3.1, we consider the sentences in the top k beams predicted by our chain extractor as input to our answer prediction model. Different datasets may require different modifications of the basic BERT model as well as different types of reasoning, so we present the answer prediction module in the following section.

Model	Oracle	Avg Length	Answer Found	Supp F1	Answer F1
Oracle	Shortest	1.6	93.6	58.5	-
Oracle	Q-Overlap	1.9	93.6	63.9	-
Oracle	Supp Facts	2.4	100.0	100.0	75.4
BERT-Para	Q-Overlap	2.0	76.3	64.5	66.0
BERT-Para	Shortest	1.5	74.1	56.8	65.5
BERT-Sent	Shortest	1.7	72.5	53.1	60.2
BiDAF-Para	Shortest	1.4	62.0	52.4	58.1
BERT-Para (top 5)	Q-Overlap	3.2	88.1	65.6	**70.3**

Table 3.1: The characteristics of different chains generated by different models under different supervision on the HotpotQA dev set: for different models and chain oracles, we report the average chain length, fraction of chains containing the answer, F1 with respect to the annotated supporting facts, and F1 on the final QA task. Here we only pick the 1-best chain in the beam.

3.4 Experimental Setup

3.4.1 Datasets

WikiHop Welbl et al. (2018) introduced this English dataset specially designed for text understanding across multiple documents. The dataset consists of around 40k questions, answers, and passages. Questions in this dataset are multiple-choice with around 10 choices on average.

HotpotQA Yang et al. (2018) proposed a new dataset with 113k English Wikipedia-based question-answer pairs. Similar to WikiHop, questions require finding and reasoning over multiple supporting documents to answer. Different from WikiHop, models should choose answers by selecting variable-length spans from these documents. Sentences relevant to finding the answer are annotated as "supporting facts" in the dataset.

3.4.2 Implementation Details

Oracle chain extraction We use the off-the-shelf NER system from AllenNLP (Gardner et al., 2018). We treat any entity that appears explicitly more than 5 times across sentences as a common entity,[3] and ignore it when we build the graph. Because these documents are only short snippets from Wikipedia, this criterion is loose enough to keep most useful mentions.

Chain extractor We use the uncased BERT tokenizer to tokenize both question and paragraphs. We use the pretrained `bert-base-uncased` model and fine-tune it using Adam with a fixed learning rate of 5e-6. At test time, we produce our chains using beam search with beam size 5.

Answer prediction We concatenate the question and the combined chains from previous step in the top k beams in the standard way as described in the original BERT paper (Devlin et al., 2019) and encode it using the pre-trained BERT model. We denote its [CLS] token as $[\text{CLS}]_p$.

WikiHop is a multiple-choice dataset. Since we need to choose an answer from a candidate list, we encode each candidate with BERT. The [CLS] token for candidate i is denoted as $[\text{CLS}]_{C_i}$. We then compute the score of a candidate C_i being choose as the dot product between $[\text{CLS}]_p$ and $[\text{CLS}]_{C_i}$.

HotpotQA is a span-based question answering task, where finding the answer requires predicting the start and end of a span in the context. We compute distributions over these positions via two learned weight matrices $\mathbf{W}_{\text{start}}$ and $\mathbf{W}_{\text{end}}$. Each position in the concatenated sequence except the [CLS] token is multiplied by

[3] These mentions are often extremely common entities like *U.S.*, which are likely to introduce spurious edges rather than good ones.

the corresponding weight matrix and softmaxed. Since we also need to predict the question type on HotpotQA (to handle yes/no questions vs. span extraction ones), we predict the type by taking the dot product of $[\text{CLS}]_p$ with a trainable matrix $\mathbf{W}_{\text{type}}$. We use `bert-large-uncased` instead of `bert-base-uncased` in the answer prediction module. We use the same optimizer and learning rate as chain extractor.

3.5 Results

In this section we aim to answer two main questions. First, which of our proposed chain extraction techniques is most effective, and how do they compare? Second, how does our approach compare to state-of-the-art models on these datasets? Finally, can we evaluate our extracted chains intrinsically: how important is *ordered* chain extraction and how well do our chains align with human intuition about question answering?

3.5.1 Comparison of Chain Extraction Methods

We first study the characteristics of our extracted chains with several experiments focused on HotpotQA. We choose this dataset since it provides human-annotated supporting facts so we can directly compare these against our model. Several statistics are shown in Table 3.1. For different combinations of our model and which choice of chain oracle we use, we calculate several statistics, as described in the caption. We have the following observations:

Using more context helps chain extractors to find relevant sentences. Comparing BERT-Para and BERT-Sent, we find that with all other parts fixed and only by encoding more context, we improve the answer prediction performance by around 5%. This may indicate that BERT can capture cross sentence relations

such as coreference and find more supporting evidence as a result. The comparison with BiDAF-Para vs. BERT-Sent also indicates this. Despite finding many fewer answer candidates (62% instead of 72%), BiDAF-Para only achieves around 2% lower performance. One possible explanation to this is that without context, the BERT extraction model may pick up "distractor" sentences related to the question but which do not actually lead to the answer, potentially confusing the answer prediction module.

The one-best chain often contains the answer. This demonstrates the effectiveness of our chain extractor: the BERT-Para model with just 2 extracted sentences can locate the answer 76% of the time. We further analyze the quality of these chains in the following sections. Note that this is nearly the same amount of evidence as in the human-labeled supporting facts (2.4 sentences on average); the difference can be explained by cases where the model can jump directly to the answer (Chen and Durrett, 2019).

Q-Overlap helps recover more supporting evidence. The main difference between our Shortest oracle and the Q-Overlap oracle is that Q-Overlap contains additional relevant sentences besides the one containing the answer. As a result, models trained with Q-Overlap should also yield a higher F1 score for finding the supporting facts, which is supported by the results (64 vs. 56).

Performance can be improved by taking a union across multiple chains In the last row, we show a version of BERT-Para where the top 5 chains in the beam have been unioned together and truncated to 5 sentences. These top 5 chains contain permutations of roughly the same sentences, so this does not greatly increase the average length. However, this greatly increases answer recall and downstream

F1. One reason is that this maintains uncertainty over the correct reasoning chain and can seamlessly handle question types involving comparison of multiple entities, which are difficult to address with a single reasoning chain of the sort presented in Figure 3.1.

3.5.2 Results compared to other systems

We evaluate our best system from the prior section (BERT-Para with top-5 chains) on the blind test sets of our two datasets. Performance is shown in Table 3.2. On WikiHop, our model significantly outperforms past published models, although these models do not use BERT. For HotpotQA, we use RoBERTa (Liu et al., 2019) weights as the pretrained model instead of BERT, which gives a performance gain. Our model achieves strong performance compared to past models, including outperforming some models which use the human-labeled supporting facts [4]

3.5.3 Evaluation of chains

Ordered extraction outperforms unordered extraction One question we can ask is how important ordered chain extraction is versus just selecting "chain-like" sentences in an unordered fashion. We compare our BERT-Para model with a variant of our model where, instead of using a pointer network to predict a chain, we make an independent classification decision for each sentence to determine whether it is relevant to the question or not. We then pick top k sentences with the highest relevance score and feed these to our BERT model. We call this model *unordered extraction*. Both are trained with the shortest-path oracle.[5] To make a fair com-

[4] This indicates that our heuristically-extracted chains can stand in effectively for this supervision, which suggests that our approach can generalize to settings where this annotation is not available.

[5] We do not use the question overlap oracle since the questions in WikiHop are synthetic like "place_of_birth gregorio di cecco", which is uninformative for the Q-overlap method.

WikiHop	dev	test
GCN (De Cao et al., 2018)	64.8	67.6
BAG (Cao et al., 2019)	66.5	69.0
CFC (Zhong et al., 2019)	66.4	70.6
JDReader (Tu et al., 2019b)	68.1	70.9
DynSAN (Zhuang and Wang, 2019)	70.1	71.4
BERT-Para (top 5)	**72.2**	**76.5**

HotpotQA	EM	F1	Supp?
DecompRC (Min et al., 2019b)	55.20	69.63	N
QFE (Nishida et al., 2019)	53.86	68.06	Y
DFGN (Qiu et al., 2019)	56.31	69.69	Y
HGN (Fang et al., 2019)	66.07	79.36	Y
SAE (Tu et al., 2019a)	66.92	79.62	Y
RoBERTa-Para (top 5)	61.20	74.11	N

Table 3.2: The blind test set performance achieved by our model on WikiHop and HotpotQA. On HotpotQA, all published works except DecompRC use the annotated supporting facts as extra supervision, which makes them not directly comparable to our model.

parison, we pick the same number of sentences ranked by prediction probability as the (top-5) chain extractor.

QA performance on those datasets is shown in Table 3.3. We also train and test our model on a hard subset of HotpotQA pointed out by (Chen and Durrett, 2019). We see that **the sequential model is more powerful than the unordered model.** On all datasets, our chain extractor leads to higher QA performance than the unordered extractor. This holds true on HotpotQA-Hard, where multi-hop reasoning is more strongly required. Even for a very powerful pre-trained model like BERT, an explicitly sequential interaction between sentences is apparently still useful for recovering related evidences. A more powerful sequential decoder may further help with the those "hard" examples. On WikiHop, the improvement yield by our chain extractor is more marginal. One reason is that correlations have

Dataset	WikiHop		HotpotQA			HotpotQA-Hard		
	Acc	%ans	F1	SP F1	%ans	F1	SP F1	%ans
Chain Extraction	72.4	72.7	69.7	63.7	90.3	56.0	59.2	78.7
Unordered Extraction	72.1	72.3	68.3	63.4	90.1	54.3	59.4	78.3

Table 3.3: The downstream QA performance of the chains generated by different models on different datasets. The performance is evaluated by accuracy and F1 score respectively in WikiHop and HotpotQA dataset.

	quite confident	somewhat confident	not confident
shortest oracle	34 / 77.7	7 / 68.6	9 / 70.6
extracted chain	37 / 81.1	7 / 64.2	6 / 50.0
annotated supporting facts	33 / 78.8	12 / 60.0	5 / 88.0

Table 3.4: The human evaluation on different evidence sets. For each row, 50 responses are bucketed based on the Turkers' confidence ratings, and numbers denote the answer F1 within that bucket.

been noted between the question and answer options (Chen and Durrett, 2019), so that the quality of the extracted evidence contributes less to the models' downstream performance.

Chain extraction is near the performance limit on HotpotQA Given our two-stage procedure, one thing we can ask is: with a "perfect" chain extractor, how well would our question answering model do? We compare the performance of the answer prediction trained with our extracted chains against that trained with the human-annotated supporting facts. As we can see in Table 3.1, BERT achieves 75.4% F1 on the annotated supporting facts, which is only 5% higher than the result achieved by our BERT-Para (top 5) extractor. A better oracle or stronger chain extractor could help close this gap, but it is already fairly small considering the headroom on the task overall. It also shows there exist other challenges to address in the question answering piece, complementary to the proposed model, like decomposing the question into different pieces (Min et al., 2019b) to further improve

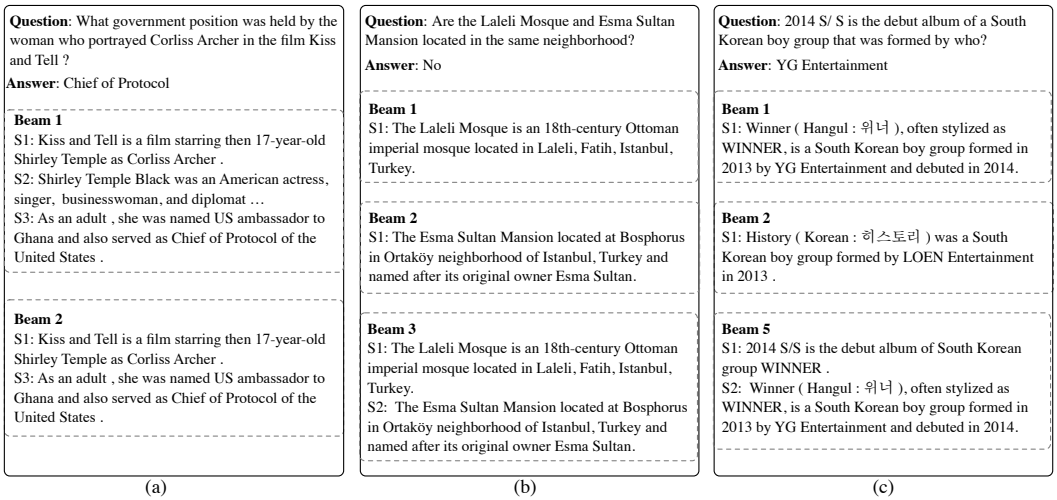


Figure 3.3: Examples of different chains picked up by our model on the development set of HotpotQA. The first shows a standard success case, the second shows success on a less common question type, and the third shows a failure case.

the multi-hop QA performance.

Human evaluation We perform a human evaluation to compare the quality of our extracted chains with our oracle as well as the annotated supporting facts. We randomly pick 50 questions in HotpotQA and ask three Turkers to answer each question based on those different evidences and rate their confidence in their answer. We pick the Turkers' answer which has the highest word overlap with the actual answer (to control for Turkers who have simply misunderstood the question) and assess their confidence in it. The statistics are shown in Table 3.4. Human performance on the three sets is quite similar – they have similar confidence in their answers, and their answers achieve similar F1 score. Although sometimes the shortest oracle may directly jump to the answer and the extracted chains may contain distractors, humans still seem to be able to reason effectively and give confidence in their answers on these short chains. Since the difference between supporting facts and our oracle on overall question answering performance is marginal, this is further

evidence that the human-annotated supporting facts may not be needed.

We also dig into the chains picked up by our chain extractor to better understand what is captured by our model. Those examples are shown in Figure 3.3. Seen from example (a), the model picks a perfect chain by first picking the sentence containing "Kiss and Tell" and "Corliss Archer", then finds the next sentence through "Shirley Temple". At the last step, it even finds a sentence via coreference. This demonstrates that although we do not explicitly model the entity links, the model still manages to learn to jump through entities in each hop.

Example (b) shows how our system can deal with comparison-style yes/no questions. There are two entities, namely, "Laleli Mosque" and "Esma Sultan Mansion" in the question, each of which must be pursued. The chain extractor proposes first a single-sentence chain about the first entity, then a single-sentence chain about the second entity. When unioned together, our answer predictor can leverage both of these together.

Example (c) shows that the extraction model picks a sentence containing the answer but without justification, it directly jumps to the answer by the lexical overlap of the two sentences and the shared entity "South Korean". The chain picked in the second beam is a distractor. There are also different distractors that contains in other hypotheses, which we do not put in the example. The fifth hypothesis contains the correct chain. This example shows that if the same entity appears multiple time in the document, the chain extractor may be distracted and pick unrelated distractors.

3.6 Related Work

Text-based multi-hop reasoning Memory Network based models (Weston et al., 2015; Sukhbaatar et al., 2015; Kumar et al., 2016; Dhingra et al., 2016; Shen et al.,

2017) try to solve multi-hop questions sequentially by using a memory cell which is designed to gather information iteratively from different parts of the passage. However, these models do not form a discrete representation of reasoning. More recent work including Entity-GCN (De Cao et al., 2018), MHQA-GRN (Song et al., 2018), and BAG (Cao et al., 2019), form this problem as a search over entity graph, and adapt graph convolution network (Kipf and Welling, 2017) to do reasoning. Such models need to construct an entity graph both at training and test time, while we only need such entities during training.

Coarse-to-fine question answering (Choi et al., 2017) combine a coarse, fast model for selecting relevant sentences and a more expensive RNN for producing the answer from those sentences. (Wang et al., 2019) apply distant supervision to generate labels and uses them to train a neural sentence extractor. Another line of work proposes to use the answer prediction score as supervision to the sentence extractor (Wang et al., 2018b; Indurthi et al., 2018; Min et al., 2018). A recent line of works on open-domain multi-hop QA (Feldman and El-Yaniv, 2019; Das et al., 2019; Qi et al., 2019; Godbole et al., 2019) also adopt the idea of forming queries in an iterative way to select the most relevant documents regarding the question. Our model differs from those works in that it operates in a more fine-grained way: it actually shows how the answer is derived rather than just retrieving relevant documents. This represents a step towards building explainable models that represent the reasoning process more explicitly (Trivedi et al., 2019; Jiang et al., 2019).

3.7 Chapter Summary

In this chapter, we learn to extract reasoning chains to answer multi-hop reasoning questions. Experimental results show that the chains are as effective as human annotations, and achieve strong performance on two large datasets. However, as remarked in past work (Chen and Durrett, 2019; Min et al., 2019a), there are several aspects of HotpotQA and WikiHop which make them require multi-hop reasoning less strongly than they otherwise might. As more challenging QA datasets are built based on lessons learned from these, we feel that reasoning in a more explicit way and properties of chain-like representations will be critical. This chapter represents the first step towards this goal of improving QA systems in such settings.

Chapter 4

Question Answering through Sub-part Alignment

This chapter is based on Chen and Durrett (2021).[1]

4.1 Introduction

As mentioned in Chapter 2, current text-based question answering models learned end-to-end often rely on spurious patterns between the question and context rather than learning the desired behavior. In this chapter, we explore a model for text-based question answering through sub-part alignment. The core idea behind our method is that if every aspect of the question is well supported by the answer context, then the answer produced should be trustable (Lewis and Fan, 2018); if not, we suspect that the model is making an incorrect prediction. The sub-parts we use are predicates and arguments from Semantic Role Labeling (Palmer et al., 2005), which we found to be a good semantic representation for the types of questions we studied. We then view the question answering procedure as a constrained graph alignment problem (Sachan and Xing, 2016), where the nodes represent the predicates and arguments and the edges are formed by relations between them (e.g. predicate-argument relations and coreference relations). Our goal is to align each

[1] Jifan Chen and Greg Durrett. 2021. Robust Question Answering Through Sub-part Alignment. In Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 1251–1263, Online. Association for Computational Linguistics.
Jifan Chen initialized the research project, conducted experiments, analyzed data and wrote the paper.

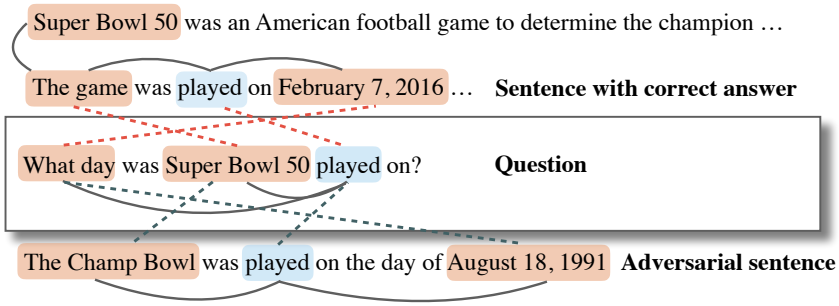


Figure 4.1: A typical example on adversarial SQuAD. By breaking the question and context down into smaller units, we can expose the incorrect entity match and use explicit constraints to fix it. The solid lines denote edges from SRL and coreference, and the dotted lines denote the possible alignments between the arguments (desired in red, actual in black).

node in the question to a counterpart in the context, respecting some loose constraints, and in the end the context node aligned to the wh-span should ideally contain the answer. Then we can use a standard QA model to extract the answer.

Figure 4.1 shows an adversarial example of SQuAD (Jia and Liang, 2017) where a standard BERT QA model predicts the wrong answer *August 18, 1991*. In order to choose the adversarial answer, our model must **explicitly** align *Super Bowl 50* to *Champ Bowl*. Even if the model still makes this mistake, this error is now exposed directly, making it easier to interpret and subsequently patch.

In our alignment model, each pair of aligned nodes is scored using BERT (Devlin et al., 2019). These alignment scores are then plugged into a beam search inference procedure to perform the constrained graph alignment. This structured alignment model can be trained as a structured support vector machine (SSVM) to minimize alignment error with heuristically-derived oracle alignments. The alignment scores are computed in a black-box way, so these individual decisions aren't easily explainable (Jain and Wallace, 2019); however, the score of an answer is directly a sum of the score of each aligned piece, making this structured prediction

phase of the model faithful by construction (Jain et al., 2020). Critically, this allows us to understand what parts of the alignment are responsible for a prediction, and if needed, constrain the behavior of the alignment to correct certain types of errors. We view this interpretability and extensibility with constraints as one of the principal advantages of our model.

We train our model on the SQuAD-1.1 dataset (Rajpurkar et al., 2016b) and evaluate on SQuAD Adversarial (Jia and Liang, 2017), Universal Triggers on SQuAD (Wallace et al., 2019), and several out-of-domain datasets from MRQA (Fisch et al., 2019). Our framework allows us to incorporate natural constraints on alignment scores to improve zero-shot performance under these distribution shifts, as well as explore coverage-accuracy tradeoffs in these settings. Finally, our model's alignments serve as "explanations" for its prediction, allowing us to ask why certain predictions are made over others and examine scores for hypothetical other answers the model could give.

5. on the first page of the chapter[s] with previously published material, please include a footnote giving the full citation of the published version[s] of [those] chapter[s] (even if they are cited elsewhere in the paper) as well as a brief statement about your personal contribution to [them]. The contribution statement might include, for example, information about your contribution to designing research, performing research, contributing new reagents or analytic tools, analyzing data, writing the dissertation or other area-specific classification of research activities.

In the following sections, we describe the details of the proposed approach and the experiments.

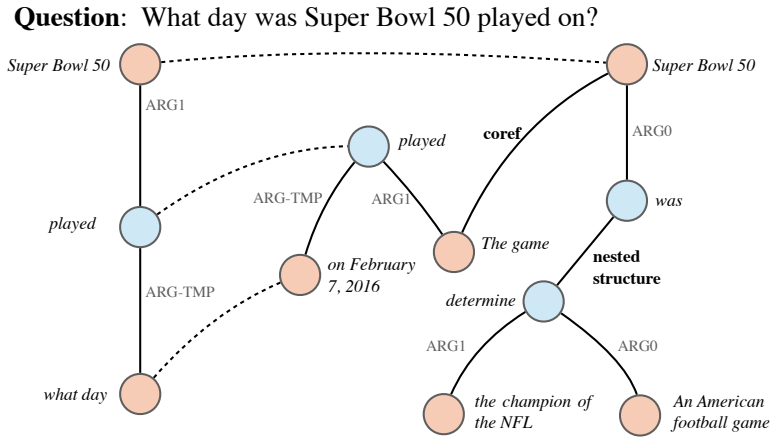


Figure 4.2: Example of our question-passage graph. Edges come from SRL, coreference (*Super Bowl 50—the game*), and postprocessing of predicates nested inside arguments (*was—determine*). The oracle alignment (Section 4.3.4) is shown with dotted lines. Blue nodes are predicates and orange ones are arguments.

4.2 QA as Graph Alignment

Our approach critically relies on the ability to decompose questions and answers into a graph over text spans. Our model can in principle work for a range of syntactic and semantic structures, including dependency parsing, SRL (Palmer et al., 2005), and AMR (Banarescu et al., 2013). We use SRL in this work and augment it with coreference links, due to the high performance and flexibility of current SRL systems (Peters et al., 2018). Throughout this work, we use the BERT-based SRL system from Shi and Lin (2019) and the SpanBERT-based coreference system from Joshi et al. (2020).

An example graph we construct is shown in Figure 4.2. Both the question and context are represented as graphs where the nodes consist of predicates and arguments. Edges are undirected and connect each predicate and its corresponding

arguments. Since SRL only captures the predicate-argument relations within one sentence, we add coreference edges as well: if two arguments are in the same coreference cluster, we add an edge between them. Finally, in certain cases involving verbal or clausal arguments, there might exist nested structures where an argument to one predicate contains a separate predicate-argument structure. In this case, we remove the larger argument and add an edge directly between the two predicates. This is shown by the edge from *was* to *determine* (labeled as *nested structure*) in Figure 4.2). Breaking down such large arguments helps avoid ambiguity during alignment.

Aligning questions and contexts has proven useful for question answering in previous work (Sachan et al., 2015; Sachan and Xing, 2016; Khashabi et al., 2018b). Our framework differs from theirs in that it incorporates a much stronger alignment model (BERT), allowing us to relax the alignment constraints and build a more flexible, higher-coverage model.

Alignment Constraints Once we have the constructed graph, we can align each node in the question to its counterpart in the context graph. In this work, we control the alignment behavior by placing explicit constraints on this process. We place a **locality constraint** on the alignment: adjacent pairs of question nodes must align no more than k nodes apart in the context. $k = 1$ means we are aligning the question to a connected sub-graph in the context, $k = \infty$ means we can align to a node anywhere in a connected component in the context graph. In our experiments, we set $k = 3$. In section 4.5, we will discuss more constraints. Altogether, these constraints define a set $\mathcal{A}$ of possible alignments.

4.3 Graph Alignment Model

4.3.1 Model

Let $\mathbf{T}$ represent the text of the context and question concatenated together. Assume a decomposed question graph $\mathbf{Q}$ with nodes $q_1, q_2, \ldots, q_m$ represented by vectors $\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_m$, and a decomposed context $\mathbf{C}$ with nodes $c_1, \ldots, c_n$ represented by vectors $\mathbf{c_1}, \ldots, \mathbf{c}_n$. Let $\mathbf{a} = (a_1, \ldots, a_m)$ be an alignment of question nodes to context nodes, where $a_i \in \{1, \ldots, n\}$ indicates the alignment of the ith question node. Each question node is aligned to exactly one context node, and multiple question nodes can align to the same context node.

We frame question answering as a maximization of an alignment scoring function over possible alignments: $\max_{\mathbf{a} \in \mathcal{A}} f(\mathbf{a}, \mathbf{Q}, \mathbf{C}, \mathbf{T})$. In this work, we simply choose f to be the sum over the scores of all alignment pairs $f(\mathbf{a}, \mathbf{Q}, \mathbf{C}, \mathbf{T}) = \sum_{i=1}^{n} S(q_i, c_{a_i}, \mathbf{T})$, where $S(q, c, \mathbf{T})$ denotes the alignment score between a question node q and a context node c. This function relies on BERT (Devlin et al., 2019) to compute embeddings of the question and context nodes and will be described more precisely in what follows. We will train this model as a structured support vector machine (SSVM), described in Section 4.3.2.

Scoring Our alignment scoring process is shown in Figure 4.3. We first concatenate the question text with the document text into $\mathbf{T}$ and then encode them using the pre-trained BERT encoder. We then compute a representation for each node in the question and context using a span extractor, which in our case is the self-attentive pooling layer of Lee et al. (2017). The node representation in the question can be computed in the same way. Then the score of a node pair is computed as a dot product $S(q, c, \mathbf{T}) = \mathbf{q} \cdot \mathbf{c}$.

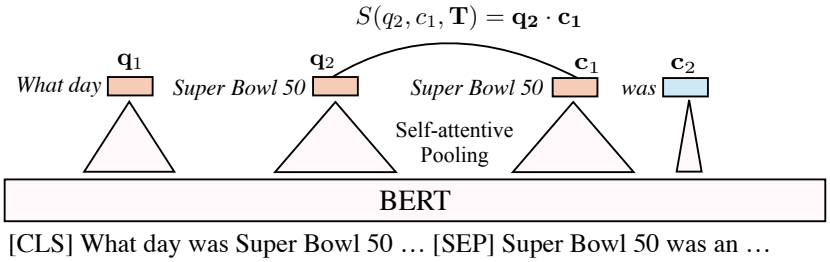


Figure 4.3: Alignment scoring. Here the alignment score is computed by the dot product between span representations of question and context nodes. The final alignment score (not shown) is the sum of these edge scores.

Answer Extraction Our model so far produces an alignment between question nodes and context nodes. We assume that one question node contains a wh-word and this node aligns to the context node containing the answer.[2] Ideally, we can use this aligned node to extract the actual answer. However, in practice, the aligned context node may only contain part of the answer and in some cases answering the question only based the aligned context node can be ambiguous. We therefore use the sentence containing the wh-aligned context node as the "new" context and use a standard BERT QA model to extract the actual answer post-hoc. In the experiments, we also show the performance of our model by only use the aligned context node without the sentence, which is only slightly worse.

4.3.2 Training

We train our model as an instance of a structured support vector machine (SSVM). Ignoring the regularization term, this objective can be viewed as a sum over the training data of a structured hinge loss with the following formulation:

[2]We discuss what to do with other questions in Section 4.4.1.

$$\sum_{i=1}^{N} \max(0, \max_{\mathbf{a} \in \mathcal{A}} [f(\mathbf{a}, \mathbf{Q}_i, \mathbf{C}_i, \mathbf{T}_i) + \text{Ham}(\mathbf{a}, \mathbf{a}_i^*)] - f(\mathbf{a}_i^*, \mathbf{Q}_i, \mathbf{C}_i, \mathbf{T}_i)])$$

where $\mathbf{a}$ denotes the predicted alignment, $\mathbf{a}_i^*$ is the oracle alignment for the ith training example, and Ham is the Hamming loss between these two. To get the predicted alignment $\mathbf{a}$ during training, we need to run loss-augmented inference as we will discuss in the next section. When computing the alignment for node j, if $a_j \neq a_j^*$, we add 1 to the alignment score to account for the loss term in the above equation. Intuitively, this objective requires the score of the gold prediction to be larger than any other hypothesis $\mathbf{a}$ by a margin of $\text{Ham}(\mathbf{a}, \mathbf{a}^*)$.

When training our system, we first do several iterations of *local training* where we treat each alignment decision as an independent prediction, imposing no constraints, and optimize log loss over this set of independent decisions. The local training helps the global training converge more quickly and achieve better performance.

4.3.3 Inference

Since our alignment constraints do not strongly restrict the space of possible alignments (e.g., by enforcing a one-to-one alignment with a connected subgraph), searching over all valid alignments is intractable. We therefore use beam search to find the approximate highest-scoring alignment: (1) Initialize the beam with top b highest aligned node pairs, where b is the beam size. (2) For each hypothesis (partial alignment) in the beam, compute a set of reachable nodes based on the currently aligned pairs under the locality constraint. (3) Extend the current hypothesis by adding each of these possible alignments in turn and accumulating its score. Beam search continues until all the nodes in the question are aligned.

An example of one step of beam hypothesis expansion with locality constraint $k = 2$ is shown in Figure 4.4. In this state, the two *played* nodes are already aligned. In any valid alignment, the neighbors of the *played* question node must be aligned within 2 nodes of the *played* context node to respect the locality constraint. We therefore only consider aligning to *the game*, *on Feb 7, 2016* and *Super Bowl 50*. The alignment scores between these reachable nodes and the remaining nodes in the question are computed and used to extend the beam hypotheses.

Note that this inference procedure allows us to easily incorporate other constraints as well. For instance, we could require a "hard" match on entity nodes, meaning that two nodes containing entities can only align if they share entities. With this constraint, as shown in the figure, *Super Bowl 50* can never be aligned to *on February 7, 2016*. We discuss such constraints more in Section 4.5.

4.3.4 Oracle Construction

Training assumes the existence of gold alignments $\mathbf{a}^*$, which must be constructed via an oracle given the ground truth answer. This process involves running inference based on heuristically computed alignment scores S_{oracle}, where $S_{\text{oracle}}(q, c)$ is computed by the Jaccard similarity between a question node q and a context node c. Instead of initializing the beam with the b best alignment pairs, we first align the wh-argument in the question with the node(s) containing the answer in the context and then initialize the beam with those alignment pairs.

If the Jaccard similarity between a question node and all other context nodes is zero, we set these as unaligned nodes. During training, our approach can gracefully handle unaligned nodes by treating these as latent variables in structured SVM: the gold "target" is then highest scoring set of alignments consistent with the gold supervision. This involves running a second decoding step on each example to

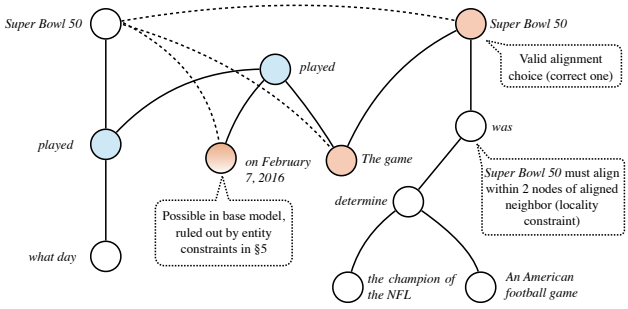


Figure 4.4: An example of constraints during beam search. The blue node *played* is already aligned. The orange nodes denote all the valid context nodes that can be aligned to for both *Super Bowl 50* and *what day* in the next step of inference given the locality constraint with $k = 2$.

impute the values of these latent variables for the gold alignment.

4.4 Experiments: Adversarial and Cross-domain Robustness

Our focus in this work is primarily robustness, interpretability, and controllability of our model. We focus on adapting to challenging settings in order to "stress test" our approach.

4.4.1 Experimental Settings

For all experiments, we train our model *only* on the English SQuAD-1.1 dataset (Rajpurkar et al., 2016b) and examine how well it can generalize to adversarial and out-of-domain settings with minimal modification, using *no fine-tuning* on new data and *no data augmentation* that would capture useful transformations. We evaluate on the `addSent` and `addOneSent` proposed by Jia and Liang (2017), and the `Universal Triggers on SQuAD` (Wallace et al., 2019). We also test the performance of our SQuAD-trained models in zero-shot adaptation to new English domains, namely Natural Questions (Kwiatkowski et al., 2019), NewsQA (Trischler et al., 2017), BioASQ (Tsatsaronis et al., 2015) and TextbookQA (Kembhavi et al.,

	SQuAD normal		SQuAD addSent		NQ		NewsQA		BioASQ		TBQA	
	ans in wh	F1	ans in wh	F1	ans in wh	F1	ans in wh	F1	ans in wh	F1	ans in wh	F1
Sub-part Alignment	84.7	84.5	49.5	**50.5**	65.8	61.5	49.3	48.1	63.5	**53.4**	35.1	**38.4**
– global train+inf	85.8	85.2	45.0	46.8	65.9	**62.3**	48.9	47.1	62.5	52.1	31.9	34.6
– ans from full sent	84.7	81.8	49.5	46.7	65.8	57.8	49.3	45.0	63.5	51.1	35.1	37.5
BERT QA	–	**87.8**	–	39.2	–	59.4	–	**48.5**	–	52.4	–	25.3

Table 4.1: The performance and ablations of our proposed model on the development sets of SQuAD, adversarial SQuAD, and four out-of-domain datasets. Our Sub-part Alignment model uses both global training and inference as discussed in Section 4.3.2-4.3.3. – **global train+inf** denotes the locally trained and evaluated model. – **ans from full sent** denotes extracting the answer using only the wh-aligned node. **ans in wh** denotes the percentage of answers found in the span aligned to the wh-span, and F1 denotes the standard QA performance measure. Here for `addSent`, we only consider the adversarial examples. Note also that this evaluation is *only on wh-questions*.

2017), taken from the MRQA shared task (Fisch et al., 2019). Our motivation here was to focus on text from a variety of domains where transferred SQuAD models may at least behave credibly. We excluded, for example, HotpotQA (Yang et al., 2018) and DROP (Dua et al., 2019), since these are so far out-of-domain from the perspective of SQuAD that we do not see them as a realistic cross-domain target.

We compare primarily against a standard **BERT QA** system (Devlin et al., 2019). We also investigate a local version of our model, where we only try to align each node in the question to its oracle, without any global training (– **global train + inf**), which can still perform reasonably because BERT embeds the whole question and context. When comparing variants of our proposed model, we only consider the questions that have a valid SRL parse and have a wh word (results in Table 4.1, Table 4.2, and Figure 4.5). When comparing with prior systems, for questions that do not have a valid SRL parse or wh word, we back off to the standard BERT QA system (results in Table 4.3).

We set the beam size $b = 20$ for the constrained alignment. We use `BERT-base-uncased`

for all of our experiments, and fine-tune the model using Adam (Kingma and Ba, 2014) with learning rate set to 2e-5. Our preprocessing uses a SpanBERT-based coreference system (Joshi et al., 2020) and a BERT-based SRL system (Shi and Lin, 2019). We limit the length of the context to 512 tokens. For our global model, we initialize the weights using a locally trained model and then fine-tune using the SSVM loss. We find the initialization helps the model converge much faster and it achieves better performance than learning from scratch. When doing inference, we set the locality constraint $k = 3$.

4.4.2 Results on Challenging Settings

The results[3] on the normal SQuAD development set and other challenging sets are shown in Table 4.1.

Our model is not as good as BERT QA on normal SQuAD but outperforms it in challenging settings. Compared to the BERT QA model, our model is fitting a different data distribution (learning a constrained structure) which makes the task harder. This kind of training scheme does cause some performance drop on normal SQuAD, but we can see that it consistently improves the F1 on the adversarial (on SQuAD addSent, a 11.3 F1 improvement over BERT QA) and cross-domain datasets except NewsQA (where it is 0.4 F1 worse). This demonstrates that learning the alignment helps improve the robustness of our model.

Global training and inference improve performance in adversarial settings, despite having no effect in-domain. Normal SQuAD is a relatively easy

[3]Here we omit `SQuAD addOneSent` for simplicity, since the performance on it has the same trend as `SQuAD addSent`. Refer to the original paper (Chen and Durrett, 2021) for the results on `SQuAD addOneSent`.

dataset and the answer for most questions can be found by simple lexical matching between the question and context. From the ablation of − **global train+inf**, we can see that more than 80% of answers can be located by matching the wh-argument. We also observe a similar pattern on Natural Questions.[4] However, as there are very strong distractors in `SQuAD addSent`, the wh-argument matching is unreliable. In such situations, the constraints imposed by other argument alignments in the question are useful to correct the wrong wh-alignment through global inference. We see that the global training plus inference is consistently better than the local version on all other datasets.

Using the strict wh answer extraction still gives strong performance From the ablation of − **ans from full sent**, we observe that our "strictest" system that extracts the answer only using the wh-aligned node is only worse by 3-4 points of F1 on most datasets. Using the full sentence gives the system more context and maximal flexibility, and allows it to go beyond the argument spans introduced by SRL. We believe that better semantic representations tailored for question answering (Lamm et al., 2020) will help further improvement in this regard.

4.4.3 Results on Universal Triggers

The results on subsets of the universal triggers dataset are shown in Table 4.2. We see that every trigger results in a bigger performance drop on BERT QA than our model. Our model is much more stable, especially on *who* and *where* question types, in which case the performance only drops by around 2%. Several factors may contribute to the stability: (1) The triggers are ungrammatical and their

[4]For the MRQA task, only the paragraph containing the short answer of NQ is provided as context, which eliminates many distractors. In such cases, those NQ questions have a similar distribution as those in SQuAD-1.1, and similarly make no use of the global alignment.

Type	Sub-part Alignment Normal	Trigger	Δ	BERT Normal	Trigger	Δ
who	84.7	82.7	2.0	87.1	78.5	8.6
why	75.1	71.3	3.8	76.5	59.7	16.8
when	88.4	82.8	5.6	90.3	80.9	9.4
where	83.6	81.4	2.2	84.1	75.8	8.3

Table 4.2: The performance of our model on the Universal Triggers on SQuAD dataset (Wallace et al., 2019). Compared with BERT, our model sees smaller performance drops on all triggers.

	Normal	addSent			addOneSent		
		overall	adv	Δ	overall	adv	Δ
R.M-Reader (Hu et al., 2018)	86.6	58.5	–	31.1	67.0	–	19.6
KAR (Wang and Jiang, 2018)	83.5	60.1	–	23.4	72.3	–	**11.2**
BERT + Adv (Yang et al., 2019b)	92.4	63.5	–	28.9	72.5	–	19.9
Our BERT	87.8	61.8	39.2	27.0	70.4	52.6	18.4
Sub-part Alignment*	84.7	**65.8**	47.1	**18.9**	**72.8**	60.1	11.9

Table 4.3: Performance of our systems compared to the literature on both `addSent` and `addOneSent`. Here, overall denotes the performance on the full adversarial set, adv denotes the performance on the adversarial samples alone. Δ represents the gap between the normal SQuAD and the overall performance on adversarial set.

arguments often contain seemingly random words, which are likely to get lower alignment scores. (2) Because our model is structured and trained to align all parts of the question, adversarial attacks on span-based question answering models may not fool our model as effectively as they do BERT.

4.4.4 Comparison to Existing Systems

In Table 4.3, we compare our best model (not using constraints from Section 4.5) with existing adversarial QA models in the literature. We note that the performance of our model on SQuAD-1.1 data is relatively lower compared to those methods, yet we achieve the best overall performance; we trade some in-distribution

performance to improve the model's robustness. We also see that our model achieves the smallest normal vs. adversarial gap on `addSent` and `addOneSent`, which demonstrates that our constrained alignment process can enhance the robustness of the model compared to prior methods like adversarial training (Yang et al., 2019b) or explicit knowledge integration (Wang and Jiang, 2018).

4.5 Generalizing by Alignment Constraints

One advantage of our explicit alignments is that we can understand and inspect the model's behavior more deeply. This structure also allows us to add constraints to our model to prohibit certain behaviors, which can be used to adapt our model to adversarial settings.

In this section, we explore how two types of constraints enable us to reject examples the model is less confident about. Hard constraints can enable us to reject questions where the model finds no admissible answers. Soft constraints allow us to set a confidence threshold for when to return our answer. We focus on evaluating our model's accuracy at various coverage points, the so-called selective question answering setting (Kamath et al., 2020).

Constraints on Entity Matches By examining `addSent` and `addOneSent`, we find the model is typically fooled when the nodes containing entities in the question align to "adversarial" entity nodes. An intuitive constraint we can place on the alignment is that we require a hard entity match—for each argument in the question, if it contains entities, it can only align to nodes in the context sharing exact the same entities.

Constraints on Alignment Scores The hard entity constraint is quite inflexible and does not generalize well, for example to questions that do not contain a entity. However, the alignment scores we get during inference time are good indicators of how well a specific node pair is aligned. For a correct alignment, every pair should get a reasonable alignment score. However, if an alignment is incorrect, there should exist some bad alignment pairs which have lower scores than the others. We can reject those samples by finding bad alignment pairs, which both improves the precision of our model and also serves as a kind of explanation as to why our model makes its predictions.

We propose to use a simple heuristic to identify the bad alignment pairs. We first find the max score $S_{\max}$ over all possible alignment pairs for a sample, then for each alignment pair (q_i, c_j) of the prediction, we calculate the worst alignment gap (WAG) $g = \min_{(q,c)\in\mathbf{a}}(S_{\max} - S(q,c))$. If g is beyond some threshold, it indicates that alignment pair is not reliable.[5]

Comparison to BERT Desai and Durrett (2020) show that pre-trained transformers like BERT are well-calibrated on a range of tasks. Since we are rejecting the unreliable predictions to improve the precision of our model, we reject the same number of examples for the baseline using the posterior probability of the BERT QA predictions. To be specific, we rank the predictions of all examples by the sum of `start` and `end` posterior probabilities and compute the F1 score on the top k predictions.

[5]The reason we look at differences from the max alignment is to calibrate the scores based on what "typical" scores look like for that instance. We find that these are on different scales across different instances, so the gap is more useful than an absolute threshold.

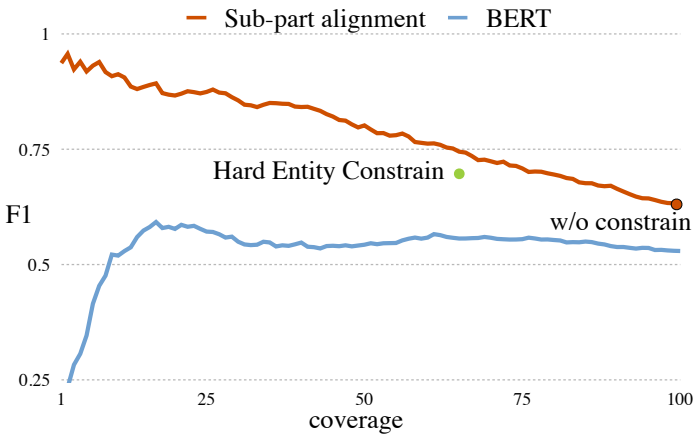


Figure 4.5: The F1-coverage curve of our model compared with BERT QA. If our model can choose to answer only the k percentage of examples it's most confident about (the coverage), what F1 does it achieve? For our model, the confidence is represented by our "worst alignment gap" (WAG) metric. Smaller WAG indicates higher confidence. For BERT, the confidence is represented by the posterior probability.

4.5.1 Results on Constrained Alignment

On Adversarial SQuAD, the confidence scores of a normal BERT QA model do not align with its performance. From Figure 4.5, we find that the highest-confidence answers from BERT (i.e., in low coverage settings) are very inaccurate. One possible explanation of this phenomenon is that BERT overfits to the pattern of lexical overlap, and is actually most confident on adversarial examples highly similar to the input. In general, BERT's confidence is not an effective heuristic for increasing accuracy.

Hard entity constraints improve the precision but are not flexible. Figure 4.5 also shows that by adding a hard entity constraint, we achieve a 71.4 F1 score which is an 8.6 improvement over the unconstrained model at a cost of only 60% of samples being covered. Under the hard entity constraint, the model is not able

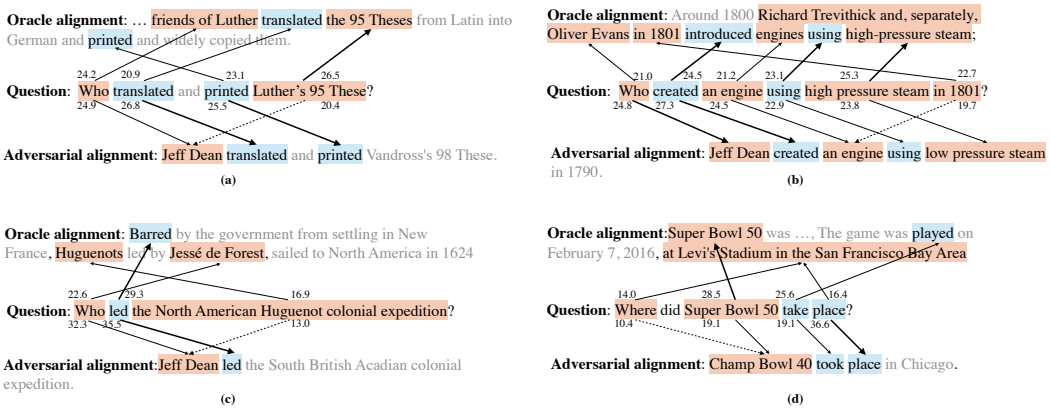


Figure 4.6: Examples of alignment of our model on `addOneSent`: both the correct alignment and also adversarial alignment are shown. The numbers are the actual alignment scores of the model's output. Dashed arrows denote the least reliable alignments and bolder arrows denote the alignment that contribute more to the model's prediction.

to align to the nodes in the adversarial sentence, but the performance is still lower than what it achieves on normal SQuAD. We examine some of the error cases and find that for a certain number of samples, there is no path from the node satisfying the constraint to the node containing the answer (e.g. they hold a more complex discourse relation while we only consider coreference as cross-sentence relation). In such cases, our method cannot find the answer.

A smaller worst alignment gap indicates better performance. As opposed to BERT, our alignment score is well calibrated on those adversarial examples. This substantiates our claim that those learned alignment scores are good indicators of how trustful alignment pairs are. Also, we see that when the coverage is the same as the entity constraint, the performance under the alignment score constraint is even better. The alignment constraints are simultaneously more flexible than the hard constraint and also more effective.

4.5.2 Case Study on Alignment Scores

In this section, we give several examples of the alignment and demonstrate how those scores can act as an explanation to the model's behavior. Those examples are shown in Figure 4.6.

As shown by the dashed arrows, all adversarial alignments contain at least one alignment with significantly lower alignment score. The model is overconfident towards the other alignments with a high lexical overlap as shown by the bold arrows. These overconfident alignments also show that the predicate alignment learned on SQuAD-1.1 is not reliable. To further improve the quality of predicate alignment, either a more powerful training set or a new predicate alignment module is needed.

Crucially, with these scores, it is easy for us to interpret our model's behavior. For instance, in example (a), the very confident predicate alignment forces *Luther's 95 Theses* to have no choice but align to *Jeff Dean*, which is unrelated. Because we have alignments over the sub-parts of a question, we can inspect our model's behavior in a way that the normal BERT QA model does not allow. We believe that this type of debuggability provides a path forward for building stronger QA systems in high-stakes settings.

4.6 Related Work

Adversarial Attacks in NLP. Adversarial attacks in NLP may take the form of adding sentences like adversarial SQuAD (Jia and Liang, 2017), universal adversarial triggers (Wallace et al., 2019), or sentence perturbations: Ribeiro et al. (2018) propose deriving transformation rules, Ebrahimi et al. (2018b) use character-level flips, and Iyyer et al. (2018) use controlled paraphrase generation. The highly structured nature of our approach makes it more robust to such attacks and provides hooks to constrain the system to improve performance further.

Neural module networks. Neural module networks are a class of models that decompose a task into several sub-tasks, addressed by independent neural modules, which make the model more robust and interpretable (Andreas et al., 2016; Hu et al., 2017; Cirik et al., 2018; Hudson and Manning, 2018; Jiang and Bansal, 2019). Like these, our model is trained end-to-end, but our approach uses structured prediction and a static network structure rather than dynamically assembling a network on the fly. Our approach could be further improved by devising additional modules with distinct parameters, particularly if these are trained on other datasets to integrate additional semantic constraints.

Unanswerable questions Our approach rejects some questions as unanswerable. This is similar to the idea of unanswerable questions in SQuAD 2.0 (Rajpurkar et al., 2018), which have been studied in other systems (Hu et al., 2019). However, techniques to reject these questions differ substantially from ours – many SQuAD 2.0 questions require not only a correct alignment between the question and context but also need to model the relationship between arguments, which is beyond the scope of this work and could be a promising future work. Also, the setting we consider here is more challenging, as we do not assume access to such questions at training time.

Graph-based QA Khashabi et al. (2018b) propose to answer questions through a similar graph alignment using a wide range of semantic abstractions of the text. Our model differs in two ways: (1) Our alignment model is trained end-to-end while their system mainly uses off-the-shelf natural language modules. (2) Our alignment is formed as node pair alignment rather than finding an optimal sub-graph, which is a much more constrained and less flexible formalism. Sachan et al. (2015); Sachan and Xing (2016) propose to use a latent alignment structure most similar to ours.

However, our model supports a more flexible alignment procedure than theirs does, and can generalize to handle a wider range of questions and datasets.

Past work has also decomposed complex questions to answer them more effectively (Talmor and Berant, 2018; Min et al., 2019b; Perez et al., 2020). Wolfson et al. (2020) further introduce a Question Decomposition Meaning Representation (QDMR) to explicitly model this process. However, the questions they answer, such as those from HotpotQA (Yang et al., 2018), are *fundamentally* designed to be multi-part and so are easily decomposed, whereas the questions we consider are not. Our model theoretically could be extended to leverage these question decomposition forms as well.

4.7 Chapter Summary

In this chapter, we presented a model for question answering through subpart alignment. By structuring our model around explicit alignment scoring, we show that our approach can generalize better to other domains. Having alignments also makes it possible to filter out bad model predictions (through score constraints) and interpret the model's behavior (by inspecting the scores).

Chapter 5

Verify QA Systems' Predictions via NLI Models

This chapter is based on Chen et al. (2021).[1]

5.1 Introduction

Recent question answering systems perform well on benchmark datasets (Seo et al., 2017; Devlin et al., 2019; Guu et al., 2020), but these models often lack the ability to verify whether an answer is correct or not; they can correctly reject some unanswerable questions (Rajpurkar et al., 2018; Kwiatkowski et al., 2019; Asai and Choi, 2021), but are not always well-calibrated to spot spurious answers under distribution shifts (Jia and Liang, 2017; Kamath et al., 2020). Natural language inference (NLI) (Dagan et al., 2005; Bowman et al., 2015) suggests one way to address this shortcoming: logical entailment provides a more rigorous notion for when a hypothesis statement is entailed by a premise statement. By viewing the answer sentence in context as the premise, paired with the question and its proposed answer as a hypothesis (see Figure 5.1), we can use NLI systems to verify that the answer proposed by a QA model satisfies the entailment criterion (Harabagiu and Hickl, 2006; Richardson et al., 2013).

[1] Jifan Chen, Eunsol Choi, and Greg Durrett. 2021. Can NLI Models Verify QA Systems' Predictions?. In Findings of the Association for Computational Linguistics: EMNLP 2021, pages 3841–3854, Punta Cana, Dominican Republic. Association for Computational Linguistics.
Jifan Chen initialized the research project, conducted experiments, analyzed data and wrote the paper.

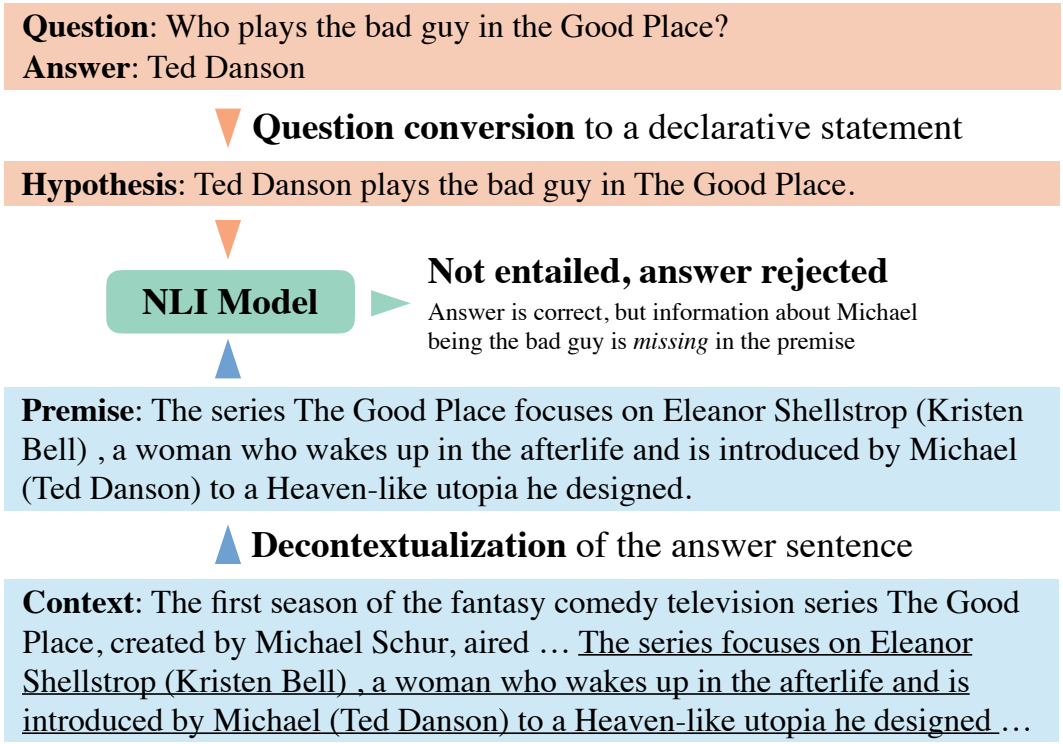


Figure 5.1: An example from the Natural Questions dataset demonstrating how to convert a (question, context, answer) triplet to a (premise, hypothesis) pair. The underlined text denotes the sentence containing the answer *Ted Danson*, which is then decontextualized by replacing *The series* with *The series The Good Place*. Although *Ted Danson* is the right answer, an NLI model determines that the hypothesis is not entailed by the premise due to missing information.

Prior work has paved the way for this application of NLI. Pieces of our pipeline like converting a question to a declarative sentence (Wang et al., 2018a; Demszky et al., 2018) and reformulating an answer sentence to stand on its own (Choi et al., 2021) have been explored. Moreover, an abundance of NLI datasets (Bowman et al., 2015; Williams et al., 2018b) and related fact verification datasets (Thorne et al., 2018) provide ample resources to train reliable models. We draw on these tools to enable NLI models to verify the answers from QA systems, and critically

investigate the benefits and pitfalls of such a formulation.

Mapping QA to NLI enables us to exploit both NLI and QA datasets for answer verification, but as Figure 5.1 shows, it relies on a pipeline for mapping a (question, answer, context) triplet to a (premise, hypothesis) NLI pair. We implement a strong pipeline here: we extract a concise yet sufficient premise through decontextualization (Choi et al., 2021), which rewrites a single sentence from a document such that it can retain the semantics when presented alone without the document. We improve a prior question conversion model (Demszky et al., 2018) with a stronger pre-trained seq2seq model, namely T5 (Raffel et al., 2020b). Our experimental results show that both steps are critical for mapping QA to NLI. Furthermore, our error analysis shows that these two steps of the process are quite reliable and only account for a small fraction of the NLI verification model's errors.

Our evaluation focuses on two factors. First, can NLI models be used to boost QA models' confidence in their decisions? Second, how does the entailment criterion of NLI, which is defined somewhat coarsely by crowd annotators (Williams et al., 2018b), transfer to QA? We train a QA model on Natural Questions (Kwiatkowski et al., 2019, NQ) and test whether using an NLI model helps it better generalize to four out-of-domain datasets from the MRQA shared task (Fisch et al., 2019). We show that by using the question converter, the decontextualization model, and the automatically generated NLI pairs from QA datasets, **our NLI model improves the prediction confidence over the base QA model across five different datasets.**[2] For example, in the selective QA setting (Kamath et al., 2020), our approach improves the F1 score of the base QA model from 81.6 to 87.1 when giving answers on the 20% of questions it is most confident about. Our pipeline

[2]The converted NLI datasets, the question converter, the decontextualizer, and the NLI model are available at `https://github.com/jifan-chen/QA-Verification-Via-NLI`

further identifies the cases where there exists an information mismatch between the premise and the hypothesis. We find that existing QA datasets encourage models to return answers when the context does not actually contain sufficient information, suggesting that fully verifying the answers is a challenging endeavor.

In the following sections, we describe the background and implementation of the proposed approach.

5.2 Using NLI as a QA Verifier

5.2.1 Background and Motivation

Using entailment for QA is an old idea; our high-level approach resembles the approach discussed in Harabagiu and Hickl (2006). Yet, the execution of this idea differs substantially as we exploit modern neural systems and newly proposed annotated data for passage and question reformulation. Richardson et al. (2013) explore a similar pipeline, but find that it works quite poorly, possibly due to the low performance of entailment systems at the time (Stern and Dagan, 2011). We believe that a combination of recent advances in natural language generation (Demszky et al., 2018; Choi et al., 2021) and strong models for NLI (Liu et al., 2019) equip us to re-evaluate this approach.

Moreover, the focus of other recent work in this space has been on transforming QA *datasets* into NLI *datasets*, which is a different end. Demszky et al. (2018) and Mishra et al. (2021) argue that QA datasets feature more diverse reasoning and can lead to stronger NLI models, particularly those better suited to strong contexts, but less attention has been paid to whether this agrees with classic definitions of entailment (Dagan et al., 2005) or short-context NLI settings (Williams et al., 2018b).

Our work particularly aims to shed light on **information sufficiency** in

question answering. Other work in this space has focused on validating answers to unanswerable questions (Rajpurkar et al., 2018; Kwiatkowski et al., 2019), but such questions may be nonsensical in context; these efforts do not address whether all aspects of a question have been covered. Methods to handle adversarial SQuAD examples (Jia and Liang, 2017) attempt to do this (Chen and Durrett, 2021), but these are again geared towards detecting specific kinds of mismatches between examples and contexts, like a changed modifier of a noun phrase. Kamath et al. (2020) frame their selective question answering techniques in terms of spotting out-of-domain questions that the model is likely to get wrong rather than more general confidence estimation. What is missing in these threads of literature is a formal criterion like entailment: when is an answer truly sufficient and when are we confident that it addresses the question?

5.2.2 Our Approach

Our pipeline consists of an answer candidate generator, a question converter, and a decontextualizer, which form the inputs to the final entailment model.

Answer Generation In this work, we focus our attention on extractive QA (Hermann et al., 2015; Rajpurkar et al., 2016a), for which we can get an answer candidate by running a pre-trained QA model.[3] We use the `Bert-joint` model proposed by Alberti et al. (2019b) for its simplicity and relatively high performance.

Question Conversion Given a question q and an answer candidate a, our goal is to convert the (q, a) pair to a declarative answer sentence d which can be treated as the hypothesis in an NLI system (Demszky et al., 2018; Khot et al., 2018). While

[3]Our approach could be adapted to multiple choice QA, in which case this step could be omitted.

rule-based approaches have long been employed for this purpose (Cucerzan and Agichtein, 2005), the work of Demszky et al. (2018) showed a benefit from more sophisticated neural modeling of the distribution $P(d \mid q, a)$. We fine-tune a seq2seq model, T5-3B (Raffel et al., 2020b), using the (a, q, d) pairs annotated by Demszky et al. (2018).

While the conversion is trivial on many examples (e.g., replacing the wh-word with the answer and inverting the wh-movement), we see improvement on challenging examples like the following NQ question: *the first vice president of India who became the president later was?* The rule-based system from Demszky et al. (2018) just replaces *who* with the answer *Venkaiah Naidu*. Our neural model successfully appends the answer to end of the question and gets the correct hypothesis.

Decontextualization Ideally, the full context containing the answer candidate could be treated as the premise to make the entailment decision. But the full context often contains many irrelevant sentences and is much longer than the premises in single-sentence NLI datasets (Williams et al., 2018b; Bowman et al., 2015). This length has several drawbacks. First, it makes transferring models from the existing datasets challenging. Second, performing inference over longer forms of text requires a multitude of additional reasoning skills like coreference resolution, event detection, and abduction (Mishra et al., 2021). Finally, the presence of extraneous information makes it harder to evaluate the entailment model's judgments for correctness; in the extreme, we might have to judge whether a fact about an entity is true based on its entire Wikipedia article, which is impractical.

We tackle this problem by *decontextualizing* the sentence containing the answer from the full context to make it stand alone. Recent work (Choi et al., 2021) proposed a sentence decontextualization task in which a sentence together with its

context are taken and the sentence is rewritten to be interpretable out of context if feasible, while preserving its meaning. This procedure can involve name completion (e.g., *Stewart* → *Kristen Stewart*), noun phrase/pronoun swap, bridging anaphora resolution, and more.

More formally, given a sentence S_a containing the answer and its corresponding context C, decontextualization learns a model $P(S_d \mid S_a, C)$, where S_d is the decontextualized form of S_a. We train a decontextualizer by fine-tuning the T5-3B model to decode S_d from a concatenation of (S_a, C) pair, following the original work. More details about the models we discuss here can be found in the original paper (Chen et al., 2021).

5.3 Experimental Settings

Our experiments seek to validate the utility of NLI for verifying answers **primarily under distribution shifts**, following recent work on selective question answering (Kamath et al., 2020). We transfer an NQ-trained QA model to a range of datasets and evaluate whether NLI improves answer confidence.

Datasets We use five English-language span-extractive QA datasets: Natural Questions (Kwiatkowski et al., 2019, NQ), TriviaQA (Joshi et al., 2017b), BioASQ (Tsatsaronis et al., 2015), Adversarial SQuAD (Jia and Liang, 2017, SQuAD-adv), and SQuAD 2.0 (Rajpurkar et al., 2018). For TriviaQA and BioASQ, we use processed versions from MRQA (Fisch et al., 2019). These datasets cover a wide range of domains including biology (BioASQ), trivia questions (TriviaQA), real user questions (NQ), and human-synthetic challenging sets (SQuAD2.0 and SQuAD-adv). For NQ, we filter out the examples in which the questions are narrative statements rather than questions by the rule-based system proposed by Demszky et al. (2018). We

also exclude the examples based on tables because they are not compatible with the task formulation of NLI.[4]

Base QA Model We train our base QA model (Alberti et al., 2019b) with the NQ dataset. To study robustness across different datasets, we fix the base QA model and investigate its capacity to transfer. We chose NQ for its high quality and the diverse topics it covers.

Base NLI Model We use the RoBERTa-based NLI model trained using Multi-Genre Natural Language Inference (Williams et al., 2018b, MNLI) from AllenNLP (Gardner et al., 2018) for its broad coverage and high accuracy.

QA-enhanced NLI Model As there might exist different reasoning patterns in the QA datasets which are not covered by the MNLI model (Mishra et al., 2021), we study whether NLI pairs *generated from QA datasets* can be used jointly with the MNLI data to improve the performance of an NLI model. To do so, we run the QA instances in the NQ training set through our QA-to-NLI conversion pipeline, resulting in a dataset we call **NQ-NLI**, containing (premise, hypothesis) pairs from NQ with binary labels. As answer candidates, we use the predictions of the base QA model. If the predicted answer is correct, we label the (premise, hypothesis) as positive (entailed), otherwise negative (not entailed). To combine NQ-NLI with MNLI, we treat the examples in MNLI labeled with "entailment" as positive and the others as negative. We take the same number of examples as of NQ-NLI from

[4]After filtering, we have 191,022/4,855 examples for the training and development sets respectively. For comparison, the original NQ contains 307,373/7,842 examples for training and development.

Question	Where was Dyrrachium located? (Answerable)	What naval base fell to the Normans? (Unanswerable)
QA Prediction	Adriatic	Dyrrachium
Hypothesis	Dyrrachium was located in Adriatic.	The naval base Dyrrachium fell to the Normans.
Premise	Dyrrachium — one of the most important naval bases of the Adriatic — fell again to Byzantine hands.	Dyrrachium — one of the most important naval bases of the Adriatic — fell again to Byzantine hands.
NLI Prediction	Entail	Not Entail

Figure 5.2: Two examples from SQuAD2.0. The MNLI model successfully accepts the correct answer for the answerable question (left) and rejects a candidate answer for the unanswerable one (right).

MNLI and shuffle them to get a mixed dataset which we call **NQ-NLI+MNLI**. We use these dataset names to indicate NLI models trained on these datasets.

The basic statistics for each dataset after processing with our pipeline can be found in the original paper (Chen et al., 2021).

5.4 Improving Selective Question Answering with NLI

In this section, we explore to what extent either off-the-shelf or QA-augmented NLI models work as verifiers across a range of QA datasets.

5.4.1 Rejecting Unanswerable Questions

We start by testing how well a pre-trained MNLI model, with an accuracy of 90.2% on held-out MNLI examples, can identify unanswerable questions in SQuAD2.0. We run our pre-trained QA model on the unanswerable questions to produce answer candidates, then convert them to the NLI pairs through our

pipeline, including question conversion and decontextualization. We run the entailment model trained on MNLI to see how frequently it is able to reject the answer by predicting either "neutral" or "contradiction". For questions with annotated answers, we also generate the NLI pairs with the gold answer and see if the entailment model trained on MNLI can accept the answer.

The MNLI model successfully rejects **78.5%** of the unanswerable examples and accepts **82.5%** of the answerable examples. Two examples taken from SQuAD2.0 are shown in Figure 5.2. We can see the MNLI model is quite sensitive to the information mismatch between the hypothesis and the premise. In the case where there is no information about *Normans* in the premise, it rejects the answer. Without seeing any data from SQuAD2.0, MNLI can already act as a strong verifier in the unanswerable setting where it is hard for a QA model to generalize (Rajpurkar et al., 2018).

5.4.2 Selective Question Answering

To analyze the effectiveness of the NLI models in a more systematic way, we test whether they can improve model performance in a "selective" QA setting (Kamath et al., 2020). That is, **if our model can choose to answer only the k percentage of examples it is most confident about (the coverage), what F1 can it achieve?** We first rank the examples by the confidence score of a model; for our base QA models, this score is the posterior probability of the answer span, and for our NLI-augmented models, it is the posterior probability associated with the "entailment" class. We then compute F1 scores at different coverage values.

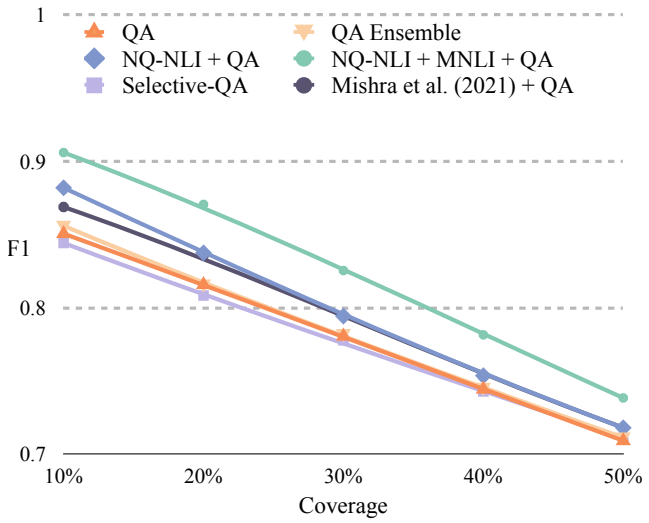


Figure 5.3: Average "selective" QA performance of our models *combining the posterior* from the NQ-NLI and the QA models over five datasets. The x-axis denotes the top k% of examples the model is answering, ranked by the confidence score. The y-axis denotes the F1 score.

5.4.2.1 Comparison Systems

NLI model variants We train separate NLI models with MNLI, NQ-NLI, NQ-NLI+MNLI introduced in Section 5.3, as well as with the NLI version of the FEVER (Thorne et al., 2018) dataset, which is retrieved by Nie et al. (2019). As suggested by Mishra et al. (2021), an NLI model could benefit from training with premises of different length; therefore, we train an NLI model without the decontextualization phase of our pipeline on the combined data from both NQ-NLI and MNLI. We call this model **Mishra et al. (2021)** since it follows their procedure. All of the models are initialized using RoBERTa-large (Liu et al., 2019) and trained using the same configurations.

NLI+QA We explore combining complementary strengths of the NLI posteriors and the base QA posteriors. We take the posterior probability of the two models as

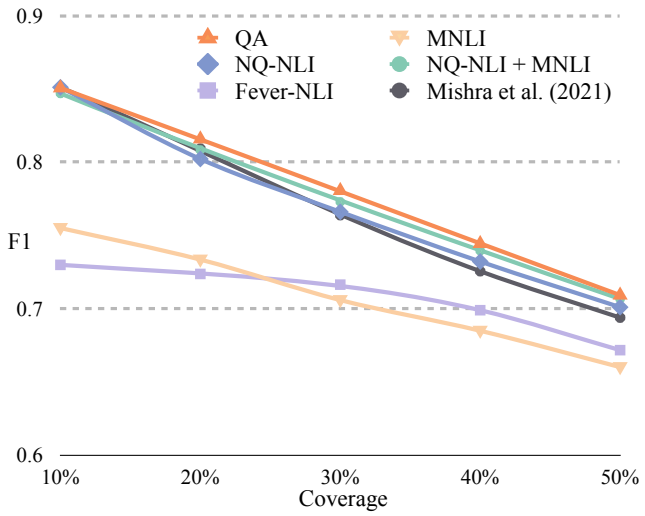


Figure 5.4: Average "selective" QA performance of our NLI models *alone* (not including QA posteriors) trained on NQ-NLI over five datasets. The x-axis denotes the top k% of examples the model is answering, ranked by the confidence score. The y-axis denotes the F1 score.

features and learn a binary classifier $y = \text{logistic}(w_1 p_{\text{QA}} + w_2 p_{\text{NLI}})$ as the combined entailment model and tune the model on 100 held-out NQ examples. **+QA** denotes this combination with any of our NLI models.

QA-Ensemble To compare with **NLI+QA**, we train another identical QA model, `Bert-joint`, using the same configurations and ensemble the two QA models using the same way as **NLI+QA**.

Selective QA Kamath et al. (2020) train a calibrator to make models better able to selectively answer questions in new domains. The calibrator is a binary classifier with seven features: passage length, the length of the predicted answer, and the top five softmax probabilities output by the QA model. We use the same configuration as (Kamath et al., 2020) and train the calibrator on the same data as our NQ-NLI

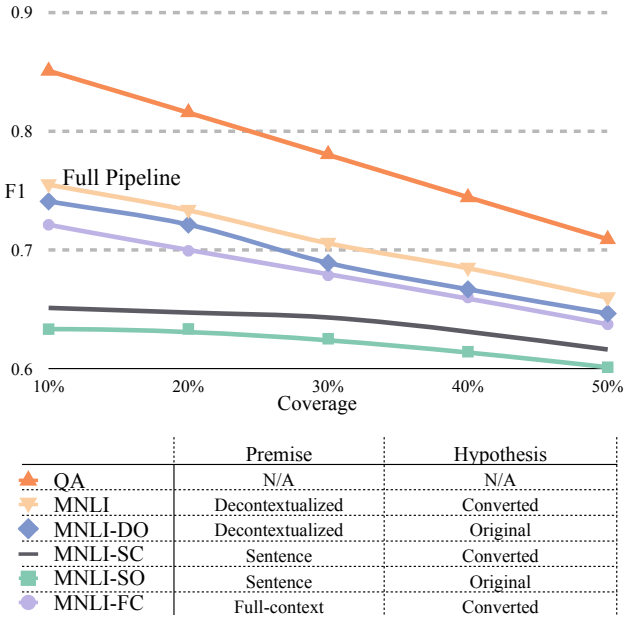


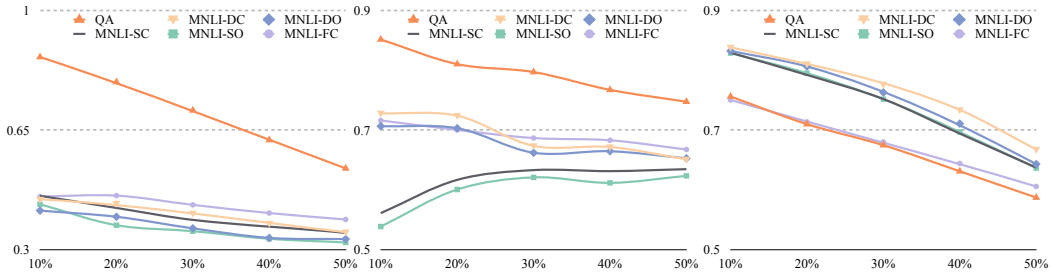
Figure 5.5: Average "selective" QA performance of the MNLI model on five QA datasets. Converted vs. original denotes using the converted question or the original question concatenated with the answer as the hypothesis. Sentence vs. decontextualized vs. full-context denotes using the sentence containing the answer, its decontextualized form, or the full context as the premise.

model.

5.4.2.2 Results and Analysis

Figure 5.3 shows the macro-averaged results over the five QA datasets. Please refer to the original paper (Chen et al., 2021) for per dataset breakdown.

Our **NQ-NLI+QA** system, which combines the QA models' posteriors with an NQ-NLI-trained system, already shows improvement over using the base QA posteriors. Surprisingly, additionally training the NLI model on MNLI (**NQ-NLI+MNLI+QA**) gives *even stronger* results. The NLI models appear to be complementary to the QA model, improving performance even on out-of-domain



(a) NQ (b) BioASQ (c) SQuAD 2.0

Figure 5.6: "Selective" QA performance of the MNLI model on three out of five QA datasets we used. Here, we omit TriviaQA and SQuAD-adv since they exhibit similar behavior as BioASQ and SQuAD2.0, respectively. The legends share the same semantics as Figure 5.5. The x-axis denotes coverage and the y-axis denotes the F1 score.

data. We also see that our our **NQ-NLI+MNLI+QA** outperforms **Mishra et al. (2021)+QA** by a large margin. By inspecting the performance breakdown, we see the gap is mainly on SQuAD2.0 and SQuAD-adv. This is because these datasets often introduce subtle mismatches by slight modification of the question or context; even if the NLI model is able to overcome other biases, these are challenging contrastive examples from the standpoint of the NLI model. This observation also indicates that to better utilize the complementary strength of MNLI, the proposed decontextualization phase in our pipeline is quite important.

Selective QA shows similar performance to using the posterior from QA model, which is the most important feature for the calibrator.

Combining NLI model with the base QA models' posteriors is necessary for this strong performance. Figure 5.4 shows the low performance achieved by the NLI models alone, indicating that **NLI models trained exclusively on NLI dataset (FEVER-NLI, MNLI) cannot be used by themselves as effective verifiers for QA**. This also indicates a possible domain or task mismatch between FEVER, MNLI, and the other QA datasets.

NQ-NLI helps bridge the gap between the QA datasets and MNLI. In Figure 5.4, both NQ-NLI and NQ-NLI+MNLI achieve similar performance to the original QA model. We also find that training using both NQ-NLI and MNLI achieves slightly better performance than training using NQ-NLI alone. This suggests that we are not simply training a QA model of a different form by using the NQ-NLI data; rather, the NQ-NLI pairs are compatible with the MNLI pairs, and the MNLI examples are useful for the model.

5.5 Effectiveness of the Proposed Pipeline

We present an ablation study on our pipeline to see how each component contributes to the final performance. For simplicity, we use the off-the-shelf MNLI model since it does not involve training using the data generated through the pipeline. Figure 5.5 shows the average results across five datasets and Figure 5.6 presents individual performance on three datasets.

We see that **both the question converter and the decontextualizer contribute to the performance of the MNLI model.** In both figures, removing either module harms the performance for all datasets. On NQ and BioASQ, using the full context is better than the decontextualized sentence, which hints that there are cases where the full context provides necessary information. We have a more comprehensive analysis in Section 5.6.2.

Moreover, we see that MNLI outperforms the base QA posteriors on SQuAD2.0 and SQuAD-adv. Figure 5.6(a) also shows that the largest gap between the QA and NLI model is on NQ, which is unsurprising since the QA model is trained on NQ. These results show how the improvement in the last section is achieved: the complementary strengths of MNLI and NQ datasets lead to the best overall performance.

5.6 Understanding the Behavior of NQ-NLI

We perform manual analysis on 300 examples drawn from NQ, TriviaQA, and SQuAD2.0 datasets where **NQ-NLI+MNLI** model produced an error. We classify errors into one of 7 classes, described in Section 5.6.1 and 5.6.2. All of the authors of this work conducted the annotation. The annotations agree with a Fleiss' kappa value of 0.78, with disagreements usually being between closely related categories among our 7 error classes, e.g., annotation error vs. span shifting, wrong context vs. insufficient context, as we will see later. The breakdown of the errors in each dataset is shown in Table 5.1.

5.6.1 Errors from the Pipeline

We see that across the three different datasets, the number of errors attributed to our pipeline approach is below 10%. This demonstrates that the question converter and the decontextualization model are quite effective to convert a (question, answer, context) triplet to a (premise, hypothesis) NLI pair. For the question converter, errors mainly happen in two scenarios as shown in Figure 5.7. (1) The question converter gives an answer of the wrong type to a question. For example, the question asks "*How old...*", but the answer returned is "*Mike Pence*" which does not fit the question. The question converter puts *Mike Pence* back into the question and yields an unrelated statement. Adding a presupposition checking stage to the question converter could further improve its performance (Kim et al., 2021). (2) The question is long and syntactically complex; the question converter just copies a long question without answer replacement.

For the decontextualization model, errors usually happen when the model fails to recall one of the required modifications. As shown in the example in Figure 5.7, the model fails to replace *The work* with its full entity name *The Art of*

Question Conversion Error
Question: How old is the vice president of the United States?
Hypothesis: Mike Pence is the vice president of the United States.

Question: Theodore Roosevelt formed the Progressive Party when he lost the Republican nomination to William Howard Taft. What was the party also known as?
Hypothesis: Theodore Roosevelt formed the Progressive Party when he lost the Republican nomination to William Howard Taft.

Decontext Error (NLI Prediction: Not Entail)
Question: Who was the author of The Art of War?
Predicted Answer / Gold Answer: Sun Tzu / Sun Tzu
Hypothesis: Sun Tzu was the author of the art of war.
Premise: The work, which is attributed to the ancient Chinese military strategist Sun Tzu ("Master Sun", also spelled Sunzi), is composed of 13 chapters.
Full Context: The Art of War is an ancient Chinese military treatise dating from the Spring and Autumn period in 5th century BC. The work, which is attributed to the ancient Chinese military strategist Sun Tzu …

Figure 5.7: Pipeline error examples from the NQ development set: the underlined text span denotes the answer predicted by the QA model.

War.

5.6.2 Errors from the NLI Model

Most of the errors are attributed to the entailment model. We investigate these cases closely and ask ourselves *if these really are errors*. We categorize them into the following categories.

Entailment These errors are truly mistakes by the entailment model: in our view, the pair of sentences should exhibit a different relationship than what was predicted.

Wrong Context The QA model gets the right answer for the wrong reason. The example in Figure 5.8 shows that *John Von Neumann* is the annotated answer but it is not entailed by the premise because no information about *CPU* is provided. Although the answer is correct, we argue it is better for the model to reject this

case. This again demonstrates one of the key advantages of using an NLI model as a verifier for QA models: it can identify cases of information mismatch like this where the model didn't retrieve suitable context to show to the user of the QA system.

Insufficient Context (out of scope for decontextualization) The premise lacks essential information that could be found in the full context, typically later in the context. In Figure 5.8, the answer *Roxette* is in the first sentence. However, we do not know that she wrote the song *It Must Have Been Love* until we go further in the context. The need to add future information is beyond the scope of the decontextualization (Choi et al., 2021).

Span Shifting The predicted answer of the QA model overlaps with the gold answer and it is acceptable as a correct answer. For example, a question asks *What Missouri town calls itself the Live Music Show Capital?* Both *Branson* and *Branson, Missouri* can be accepted as the right answer.

Annotation Error Introduced by the incomplete or wrong annotations – some acceptable answers are missing or the annotated answer is wrong.

From Table 5.1, we see that "wrong context" cases consist of 25% and 40% of the errors for NQ and TriviaQA, respectively, while they rarely happen on SQuAD2.0. This is because the supporting snippets for NQ and TriviaQA are retrieved from Wikipedia and web documents, so the information contained may not be sufficient to support the question. For SQuAD2.0, the supporting document is given to the annotators, so no such errors happen.

This observation indicates that the NLI model can be particularly useful in the open-domain setting where it can reject answers that are not well supported. In

	NQ		TQA		SQuAD2.0	
Question Conversion	3	0	0	2	2	0
Decontext	0	4	0	0	0	7
Entailment	12	39	2	14	12	56
Wrong Context	0	23	0	42	0	2
Insufficient Context	0	11	0	16	0	4
Span Shifting	3	0	13	0	7	0
Annotation	5	0	11	0	10	0
Total	23	77	26	74	31	69

Table 5.1: Error breakdown of our **NQ-NLI+MNLI** verifier on NQ, TQA (TriviaQA), and SQuAD2.0. Here, yellow and purple denote the false positive and false negative counts respectively. False positive: NLI predicts entailment while the answer predicted is wrong. False negative: NLI predicts non-entailment while the answer predicted is right.

particular, we believe that this raises a question about answers in TriviaQA. The supporting evidence for the answer is often **insufficient** to validate all aspects of the question. **What should a QA model do in this case: make an educated guess based on partial evidence, or reject the answer outright?** This choice is application-specific, but our approach can help system designers make these decisions explicit.

Around 10% to 15% of errors happens due to insufficient context. Such errors could be potentially fixed in future work by learning a question-conditioned decontextualizer which aims to gather all information related to the question.

5.7 Related Work

NLI for Downstream Tasks Welleck et al. (2019) proposed a dialogue-based NLI dataset and the NLI model trained over it improved the consistency of a dialogue system; Pasunuru et al. (2017); Li et al. (2018); Falke et al. (2019) used NLI models to detect factual errors in abstractive summaries. For question answering,

Entailment Error (NLI Prediction: Not Entail)
Question: What were the results of the development of Florida's railroads?
Predicted / Gold Answer: towns grew and farmland was cultivated / towns grew and farmland was cultivated
Hypothesis: The results of the development of Florida's railroads were that towns grew and farmland was cultivated.
Premise: Henry Flagler built a railroad along the east coast of Florida and eventually to Key West; towns grew and farmland was cultivated along the rail line.

Entailment Error (NLI Prediction: Entail)
Question: who is darrell brother in The Walking Dead?
Predicted / Gold Answer: Daryl / Merle Dixon
Hypothesis: Daryl is darrell brother in the walking dead.
Premise: The character Merle Dixon was first introduced in the first season of The Walking Dead as a Southern redneck hunter who has a younger brother, Daryl

Wrong Context Error (NLI Prediction: Not Entail)
Question: Who developed the central processing unit (cpu)?
Predicted Answer / Gold Answer: Jonh von Neumann / Jonh von Neumann
Hypothesis: John von Neumann developed the central processing unit (cpu).
Premise: On June 30, 1945, before ENIAC was made, mathematician John von Neumann distributed the paper entitled First Draft of a Report on the EDVAC.

Insufficient Context Error (NLI Prediction: Not Entail)
Question: Who sang It Must Have Been Love?
Predicted Answer / Gold Answer: Roxette / Roxette
Hypothesis: Roxette sang it must have been love.
Premise: Roxette are a Swedish pop rock duo, consisting of Marie Fredriksson and Per Gessle.
Full Context: Roxette are a Swedish pop rock duo, consisting of Marie Fredriksson and Per Gessle … She went on to achieve nineteen UK Top 40 hits and several US Hot 100 hits, including four US number-ones with "The Look," "Listen to Your Heart," "It Must Have Been Love,"…

Figure 5.8: Examples taken from the development sets of NQ and TriviaQA, grouped by different types of errors the entailment model makes. The underlined text span denotes the answer predicted by the QA model. The yellow box denotes a false positive example and the purple box denotes false negative examples.

Harabagiu and Hickl (2006) showed that textual entailment can be used to enhance the accuracy of the open-domain QA systems; Trivedi et al. (2019) used a pretrained NLI model to select relevant sentences for multi-hop question answering; Yin et al. (2020) tested whether NLI models generalize to QA setting in a few-shot learning scenario.

Our work is most relevant to Mishra et al. (2021); they also learn an NLI model using examples generated from QA datasets. Our work differs from theirs in a few chief ways. First, we improve the conversion pipeline significantly with decontextualization and a better question converter. Second, we use this framework

to improve QA performance by using NLI as a verifier, which is only possible because the decontextualization allows us to focus on a single sentence. We also study whether the converted dataset is compatible with other off-the-shelf NLI datasets. By contrast, Mishra et al. (2021) use their converted NLI dataset to aid other tasks such as fact-checking. Finally, the contrast we establish here allows us to conduct a thorough human analysis over the converted NLI data and show how the task specifications of NLI and QA are different (Section 5.6.2).

Robust Question Answering Modern QA systems often give incorrect answers in challenging settings that require generalization (Rajpurkar et al., 2018; Chen and Durrett, 2019; Wallace et al., 2019; Gardner et al., 2020; Kaushik et al., 2019). Models focusing on robustness and generalizability have been proposed in recent years: Wang and Bansal (2018); Khashabi et al. (2020); Liu et al. (2020) use perturbation based methods and adversarial training; Lewis and Fan (2018) propose generative QA to prevent the model from overfitting to simple patterns; Yeh and Chen (2019); Zhou et al. (2020) use advanced regularizers; Clark et al. (2019) debias the training set through ensemble-based training; and Chen and Durrett (2021) incorporate an explicit graph alignment procedure.

Another line of work to make models more robust is by introducing answer verification (Hu et al., 2019; Kamath et al., 2020; Wang et al., 2020b; Zhang et al., 2021) as a final step for question answering models. Our work is in the same vein, but has certain advantages from using an NLI model. First, the answer verification process is more explicit so that one is able to spot where the error emerges. Second, we can incorporate NLI datasets from other domains into the training of our verifier, reducing reliance on in-domain labeled QA data.

5.8 Chapter Summary

This chapter presents a strong pipeline for converting QA examples into NLI examples, to verify the answer with NLI predictions. The answer to the question posed in the title is **yes** (NLI models can validate these examples), with two caveats. First, it is helpful to create QA-specific data for the NLI model. Second, the information that is sufficient for a question to be fully answered may not align with annotations in the QA dataset. We encourage further explorations of the interplay between these tasks and careful analysis of the predictions of QA models.

Chapter 6

Generating Literal and Implied Subquestions to Fact-check Complex Claims

This chapter is based on Chen et al. (2022).[1]

6.1 Introduction

Fact-checking process can be decomposed to four stages (Vlachos and Riedel, 2014a): (1) extract statements to be fact-checked; (2) construct appropriate questions; (3) obtain the answers from relevant sources; (4) reach a verdict using these answers. In this Chapter, we study fact-checking as modularized question answering.

Despite a flurry of recent research on automated fact-checking Wang (2017); Rashkin et al. (2017); Volkova et al. (2017); Ferreira and Vlachos (2016); Popat et al. (2017); Tschiatschek et al. (2018), we remain far from building reliable fact-checking systems Nakov et al. (2021). This challenge motivated us to build more explainable models so the explanations can at least help a user interpret the results Atanasova et al. (2020). However, such purely extractive explanations do not necessarily help users interpret a model's reasoning process. An ideal explanation should do what a human-written fact-check does: systematically dissect different parts of the claim

[1] Jifan Chen, Aniruddh Sriram, Eunsol Choi, and Greg Durrett. 2022. Generating Literal and Implied Subquestions to Fact-check Complex Claims. In Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 3495–3516, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
Jifan Chen initialized the research project, conducted experiments, analyzed data and wrote the paper.

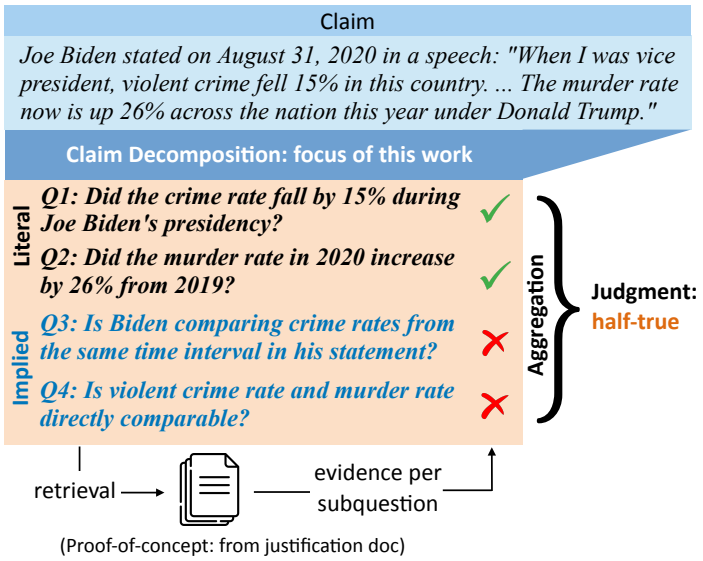


Figure 6.1: An example claim decomposition: the top two subquestions follow explicitly from the claim and the bottom two represent implicit reasoning needed to verify the claim. We can use the decomposed questions to retrieve relevant evidence (Section 6.6), and aggregate the decisions of the sub-questions to derive the final veracity of the claim (Section 6.5.3).

and evaluate their veracity.

We take a step towards explainable fact-checking with a new approach and accompanying dataset, ClaimDecomp, of decomposed claims from PolitiFact. Annotators are presented with a claim *and* the justification paragraph written by expert fact-checkers, from which they annotate a set of yes-no subquestions that give rise to the justification. These subquestions involve checking both the explicit and implicit aspects of the claim (Figure 6.1).

Such a decomposition can play an important role in an interpretable fact verification system. First, the subquestions provide a comprehensive explanation of how the decision is made: in Figure 6.1, although the individual statistics mentioned by Biden are correct, they are from different time intervals and not directly

Claim: A Facebook post stated on January 31, 2021: "Nancy Pelosi bought $1.25 million in Tesla stock the day before Joe Biden signed an order "for all federal vehicles" to be electric."

Justification: An image shared on Facebook claims that Nancy Pelosi bought $1.25 million in Tesla stock the day before Biden signed an order for all federal vehicles to be electric, implying that she sought to profit from inside information about new government policies. The House speaker did report transactions involving Tesla stock, but the post misrepresented the purchases and Biden's policies to create the false impression that the transactions represented improper insider trading in Tesla shares.

Annotation: Question	Answer	Question Source	
Were the stock purchases improper insider trading?	No	Claim○	Justification●
Does the executive order Biden signed require all federal vehicles to be electric?	Unknown	Claim●	Justification○
Did Nancy Pelosi buy 1.25 million Tesla stock the day before Joe Biden signed an order about electric vehicles?	Unknown	Claim●	Justification○

Figure 6.2: An example of our annotation process. The annotators are instructed to write a set of subquestions, give binary answers to them, and attribute them to a source. If the answer cannot be decided from the justification paragraph, "Unknown" is also an option. The question is either based on the claim or justification, and the annotators also select the relevant parts (color-coded in the figure) on which the question is based.

comparable, which yields the final judgment of the claim as "half-true". We can estimate the veracity of a claim using the decisions of the subquestions (Section 6.5.3). Second, we show that decomposed subquestions allow us to retrieve more relevant paragraphs from the verification document than using the claim alone (Section 6.6), since some of the subquestions tackle implicit aspects of a claim. We do not build a full pipeline for fact verification in this chapter, as there are other significant challenges this poses, including information which is not available online or which needs to be parsed out of statistical tables Singh et al. (2021a). Instead, we focus on showing how these decomposed questions can fit into a fact-checking pipeline through a series of proof-of-concept experiments.

Equipped with ClaimDecomp dataset, we train a model to generate decompositions of complex political claims. We experiment with pre-trained sequence-to-sequence models Raffel et al. (2020b), generating either a sequence of questions or a single question using nucleus sampling (Holtzman et al., 2020) over multiple rounds. This model can recover 58% of the subquestions, including some implicit subques-

Split	# unique claims	# tokens per claim	avg. # subquestions in single annotation	Answer %			Source %	
				Yes	No	Unknown	Justification	Claim
Train	800	33.4	2.7	48.9	45.3	5.8	83.6	16.4
Validation	200	33.8	2.7	48.3	44.8	6.9	79.0	21.0
Validation-sub	50	33.7	2.9	45.2	47.8	7.0	90.4	9.6
Test	200	33.2	2.7	45.8	43.1	11.1	92.1	7.9

Table 6.1: Statistics of the ClaimDecomp dataset. Each claim is annotated by two annotators, yielding a total of 6,555 subquestions. The second column blocks (Answer % and Source %) report the statistics at the subquestion level; Source % denotes the percentage of subquestions based on the text from the justification or the claim.

tions. To summarize, we show that decomposing complex claims into subquestions can be learned with our dataset, and reasoning with such subquestions can lead improve evidence retrieval and judging the veracity of the whole claim.

In the following sections, we elaborate on the motivation, task setup, and experiments.

6.2 Motivation and Task

Facing the complexities of real-world political claims, simply giving a final veracity to a claim often fails to be persuasive (Guo et al., 2022). To make the judgment of an automatic fact-checking system understandable, most previous work has focused on generating *justifications* for models' decisions. Popat et al. (2018); Shu et al. (2019); Lu and Li (2020) used attention weights of the models to highlight the most relevant parts of the evidence, but these only deal with explicit propositions of a claim. Ahmadi et al. (2019); Gad-Elrab et al. (2019) used logic-based systems to generate justifications, yet the systems are often based on existing knowledge graphs and are hard to adapt to complex real-world claims. Atanasova et al. (2020) treated the justification generation as a summarization problem in which they generate a justification paragraph according to some relevant evidence. Even so, it is hard to

know which parts of the claim are true and which are not, and how the generated paragraph relates to the veracity.

What is missing in the literature is a better intermediate representation of the claim: with more complex claims, explaining the veracity of a whole claim at once becomes more challenging. Therefore, we focus on decomposing the claim into a **minimal** yet **comprehensive** set of yes-no subquestions, whose answers can be aggregated into an inherently explainable decision. As the decisions to the subquestions are explicit, it is easier for one to spot the discrepancies between the veracity and the intermediate decisions.

Claims and Justifications Our decomposition process is inspired by fact checking documents written by professional fact checkers. In the data we use from PolitiFact, each **claim** is paired with a **justification paragraph** (see Figure 6.2) which contains the most important factors on which the veracity made by the fact-checkers is based. Understanding *what questions are answered in this paragraph* will be the core task our annotators will undertake to create our dataset. However, we frame the claim decomposition task (in the next section) without regard to this justification document, as it is not available at test time.

Claim Decomposition Task We define the task of complex claim decomposition. Given a claim c and the context o of the claim (speaker, date, venue of the claim), the goal is to generate a set of N yes-no subquestions $\mathbf{q} = \{q_1, q_2, ...q_N\}$. The **set** of subquestions should have the following properties:

- **Comprehensiveness:** The questions should cover as many aspects of the claim as possible: the questions should be sufficient for someone to judge the veracity of the claim.

- **Conciseness:** The question set should be as minimal as is practical and not contain repeated questions asking about minor, correlated variants seeking the same information.

An individual subquestion should also exhibit:

- **Relevance:** The answer to subquestion should help a reader determine the veracity of the claim. Knowing an answer to a subquestion should change the reader's belief about the veracity of the original claim (Section 6.5.3).
- **Fluency / Clarity:** Each subquestion should be clear, fluent, and grammatically correct (Section 6.3).

We do not require subquestions to stand alone (Choi et al., 2021); they are instead interpreted with respect to the claim and its context.

Evaluation Metric We set the model to generate the target number of subquestions, which matches the number of subquestions in the reference, guaranteeing a concise subquestion set. Thus, we focus on measuring the other properties with reference-based evaluation. Specifically, given an annotated set of subquestions and an automatically predicted set of subquestions, we assess **recall**: how many subquestions in the reference set are covered by the generated question set? A subquestion in the reference set is considered as being recalled if it is **semantically equivalent** to one of the generated subquestions by models.[2] Our notion of equivalence is nuanced and contextual: for example, the following two subquestions are considered

[2]There are cases where one generated question covers several reference questions, e.g., treating the whole claim as a question, in which case we only consider one of the reference questions to be recalled.

semantically equivalent: *"Is voting in person more secure than voting by mail?"* and *"Is there a greater risk of voting fraud with mail-in ballots?"*. We manually judge the question equivalence, as our experiments with automatic evaluation metrics did not yield reliable results (details in the original paper (Chen et al., 2022)).

6.3 Dataset Collection

Claim / Verification Document Collection We collect political claims and corresponding verification articles from PolitiFact.[3] Each article contains one justification paragraph (see Figure 6.2) which states the most important factors on which the veracity made by the fact-checkers is based. Understanding what questions are answered in this paragraph will be the core annotation task. Each claim is classified as one of six labels: *pants on fire* (most false), *false*, *barely true*, *half-true*, *mostly true*, and *true*. We collect the claims from top 50 PolitiFact pages for each label, resulting in a total of 6,859 claims.

A claim like *"Approximately 60,000 Canadians currently live undocumented in the USA."* hinges on checking a single statistic and is less likely to contain information beyond the surface form. Therefore, we mainly focus on studying complex claims in this chapter. To focus on complex claims, we filter claims with 3 or fewer verbs. We also filter out claims that do not have an associated justification paragraph. After the filtering, we get a subset consisting 1,494 complex claims.

Decomposition Annotation Process Given a claim paired with the justification written by the professional fact-checker on PolitiFact, we ask our annotators to reverse engineer the fact-checking process: generate yes-no questions which are

[3] https://www.politifact.com/

	ALL QS	MORE QS	FEWER QS
% of unmatched Qs	18.4	26.1	8.5

Table 6.2: Inter-annotator agreement assessed by the percentage of questions for which the semantics cannot be matched to the other annotator's set. We name the question set containing more questions as MORE QS and the other one as LESS QS. ALL QS is the average of MORE QS and LESS QS.

answered in the justification. As shown in Figure 6.2, for each question, the annotators also (1) give the answer; (2) select the relevant text in the justification or claim that is used for the generation (if any). The annotators are instructed to cover as many of the assertions made in the claim as possible without being overly specific in their questions.

This process gives rise to both **literal questions**, which follow directly from the claim, and **implied questions**, which are not necessarily as easy to predict from the claim itself. These are not attributes labeled by the annotators, but instead labels the authors assign post-hoc (described in Section 6.5).

We recruit 8 workers with experience in literature or politics from the freelancing platform Upwork to conduct the annotation. The original paper (Chen et al., 2022) includes details about the hiring process, workflow, as well as instructions and the UI.

Dataset statistics and inter-annotator agreement Table 6.1 shows the statistics of our dataset. We collect two sets of annotations per claim to improve subquestion coverage. We collect a total of 6,555 subquestions for 1,200 claims. Most of the questions arise from the justification and most of the questions can be answered by the justification. In addition, we randomly sample 50 claims from the validation set for our human evaluation in the rest of this chapter. We name this

set **Validation-sub**.

Comparing sets of subquestions from different annotators is nontrivial: two annotators may choose different phrasings of individual questions and even different decompositions of the same claim that end up targeting the same pieces of information. Thus, we (the authors) manually compare two sets of annotations to judge inter-annotator agreement: given two sets of subquestions on the same claim, the task is to identify questions for which the semantics are not expressed by the other question *set*. If no questions are selected, it means that the two annotators show strong agreement on what should be captured in subquestions. Example annotations are shown in the original paper (Chen et al., 2022).

We randomly sample 50 claims from our dataset and three of the authors conduct the annotation. The authors agree on this comparison task reasonably, with a Fleiss' Kappa (Fleiss, 1971) value of 0.52. The comparison results are shown in Table 6.2. On average, the semantics of 18.4% questions are not expressed by the other set. This demonstrates the **comprehensiveness** of our set of questions: only a small fraction is not captured by the other set, indicating that independent annotators are not easily coming up with distinct sets of questions. Because most questions are covered in the other set, we view the agreement as high. A simple heuristic to improve comprehensiveness further is to prefer the annotator who annotated more questions. If we consider the fraction of unmatched questions in the FEWER QS, we see this drops to 8.5%.[4] Through this manual examination, we also found that annotated questions are overall concise, fluent, clear, and grammatical.

[4]Merging two annotations results in many duplicate questions and deduplicating these without another round of adjudication is cognitively intensive. We opted not to do this due to the effectiveness of simply taking the larger set of questions.

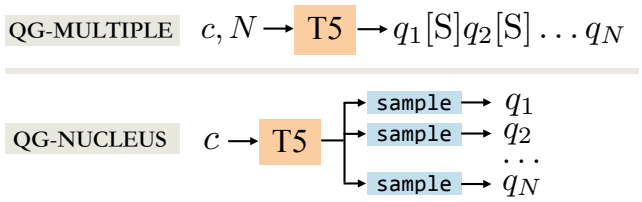


Figure 6.3: Illustration of our two question generators. QG-MULTIPLE generates all questions as a sequence while QG-NUCLEUS generates one question at a time through multiple samples.

Model	R-all	R-literal	R-implied
QG-MULTIPLE	0.58	0.74	0.18
QG-NUCLEUS	0.43	0.59	0.11
QG-MULTIPLE-JUSTIFY	0.81	0.95	0.50
QG-NUCLEUS-JUSTIFY	0.52	0.72	0.18

Table 6.3: Human evaluation results on the Validation-sub set (N=146). R-all denotes the recall for all questions; R-literal and R-implied denotes the recall for the literal questions and the implied questions respectively.

6.4 Automatic Claim Decomposition

The goal is to generate a subquestion set $\mathbf{q}$ from the input claim c, the context o, and the target number of subquestions k.

Models We fine-tune a T5-3B (Raffel et al., 2020b) model to automate the question generation process under two settings: QG-MULTIPLE and QG-NUCLEUS as shown in Figure 6.3. Both generation methods generate the same number of subquestions, equal to the number of subquestions generated by an annotator.

qg-multiple We learn a model $P(\mathbf{q} \mid c, o)$ to place a distribution over sets of subquestions given the claim and output. The annotated questions are concatenated by their annotation order to construct the output.

Question Type	# Questions	R1-P	R2-P	RL-P
Literal	2.15	0.56	0.30	0.47
Implied	1.02	0.28	0.09	0.22

Table 6.4: Number of questions of each type per claim and their lexical overlap with the claim measured by ROUGE-1, ROUGE-2, and ROUGE-L precision (how many n-grams in the question are also in the claim).

Domain knowledge (38.8%)	**Claim**: "When President Obama was elected, the market crashed … Trump was up 9%, President Obama was down 14.8% and President Bush was down almost 4%. There is an instant reaction on Wall Street." **Question**: Did Obama cause the stock market crash when he was elected? (**Domain knowledge of whether the stock market is correlated with the election.**)
Context (37.6%)	**Claim**: With voting by mail, "you get thousands and thousands of people … signing ballots all over the place." **Question**: Is there a greater risk of voting fraud with mail-in ballots? (**Need to know the background that the claim is about the potential risks of mail-in ballots.**)
Implicit meaning (16.5%)	**Claim**: Nancy Pelosi bought $1.25 million in Tesla stock the day before Joe Biden signed an order "for all federal vehicles" to be electric. **Question**: Were the stock purchases improper insider trading? **(The claim implies this purchase is insider trading.)**
Statistical rigor (7.1%)	**Claim:** "No other country witnesses the number of gun deaths that we do here in the U.S., and it's not even close." **Question:** Is the United States the country with the the highest percentage of gun deaths? (**Highest number of gun deaths does not entail highest percentage of gun deaths.)**

Figure 6.4: Four types of reasoning needed to address subquestions with their proportion (left column) and examples (right column). It shows that a high proportion of the questions need either domain knowledge or related context.

qg-nucleus We learn a model $P(q \mid c, o)$ to place a distribution over single subquestions given the claim and output. For training, each annotated subquestion is paired with the claim to form a *distinct* input-output pair. At inference, we use nucleus sampling to generate questions. See the original paper (Chen et al., 2022) for training details.

We also train these generators in an oracle setting where the justification paragraph is appended to the claim to understand how well the question generator does with more information. We denote the two oracle models as QG-MULTIPLE-VERIFY and QG-NUCLEUS-VERIFY respectively.

Results All models are trained on the training portion of our dataset and evaluated on the Validation-sub set. One of the authors evaluated the recall of each annotated subquestion in the generated subquestion set. The results are shown in Table 6.3. We observe that **most of the literal questions can be generated while only a few of the implied questions can be recovered.** Generating multiple questions as a single sequence (QG-MULTIPLE) is more effective than sampling multiple questions (QG-NUCLEUS). Many questions generated from QG-NUCLEUS are often slightly different but share the same semantics. We see that more than 70% of the literal questions and 18% of the implied questions can be generated by the best QG-MULTIPLE model. By examining the generated implied questions, we find that most of them belong to the **domain knowledge** category in Section 6.5.

Some questions could be better generated if related evidence were retrieved first, especially for questions of the **context** category (Section 6.5). The QG-MULTIPLE-JUSTIFY model can recover most of the literal questions and half of the implied questions. Although this is an oracle setting, it shows that when given proper information about the claim, the T5 model can achieve much better performance.

Qualitative Analysis While our annotated subquestion sets cover most relevant aspects of the claim, we find some generated questions are good subquestions that are missing in our annotated set, though less important. For example, for our introduction example shown in Figure 6.1, the QG-NUCLEUS model generates the question "*Is Trump responsible for the increased murder rate?*" Using the question generation model in collaboration with humans might be a promising direction for more comprehensive claim decomposition. See the original paper (Chen et al., 2022) for more examples.

6.5 Analyzing Decomposition Annotations

In this section, we study the characteristics of the annotated questions. We aim to answer: (1) How many of the questions address implicit facets of the claim, and what are the characteristics of these? (2) How do our questions differ from previous work on question generation for fact checking (Fan et al., 2020)? (3) Can we aggregate subquestion judgments for the final claim judgment?

6.5.1 Subquestion Type Analysis

We (the authors) manually categorize 285 subquestions from 100 claims in the development set into two disjoint sets: *literal* and *implied*, where *literal* questions are derived from the surface information of the claim – whether a question can be posed by only given the claim, and *implied* questions are those that need extra knowledge in order to pose.

Table 6.4 shows basic statistics about these sets, including the average number of subquestions for each claim and lexical overlap between subquestions and the base claims, evaluated with ROUGE precision, as one subquestion can be a subsequence of the original claim. On average, each claim contains one implied question which represents the deeper meaning of the claim. These implied questions overlap less with the claim.

We further manually categorize the implied questions into the following four categories, reflecting what kind of knowledge is needed to pose them (examples in Figure 6.4). Two authors conduct the analysis over 50 examples and the annotations agree with a Cohen's Kappa (Cohen, 1960) score of 0.74.

Domain knowledge The subquestion seeks domain-specific knowledge, for example asking about further steps of a legal or political process.

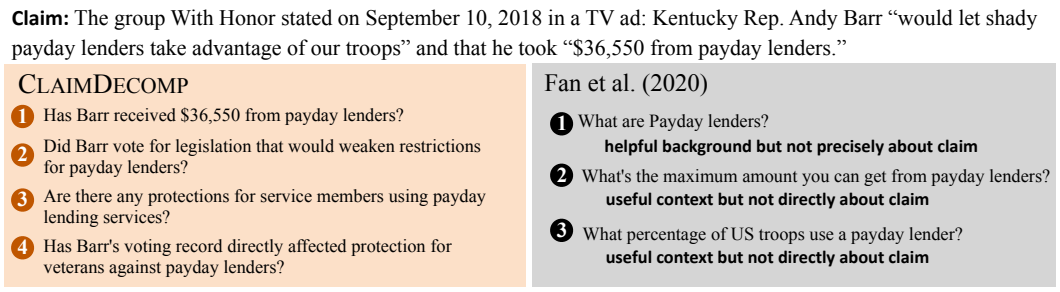


Figure 6.5: Comparison between our decomposed questions with QABriefs (Fan et al., 2020). In general, our decomposed questions are more comprehensive and relevant to the original claim.

Context The subquestion involves knowing that broader context is relevant, such as whether something is broadly common or the background of the claim (political affiliation of the politician, history of the events stated in the claim, etc).

Implicit meaning The subquestion involves unpacking the implicit meaning of the claim, specifically anchored to what the speaker's intent was.

Statistical rigor The subquestion involves checking over-claimed or over-generalized statistics (e.g., the highest raw count is not the highest per capita).

Most of the implied subquestions require either domain knowledge or context about the claim, reflecting the challenges behind automatically generating such questions.

6.5.2 Comparison to QABriefs

Our work is closely related to the QABriefs dataset (Fan et al., 2020), where they also ask annotators to write questions to reconstruct the process taken by professional fact-checkers provided the claim and its verification document.

While sharing similar motivation, we use a significantly different annotation process than theirs, resulting in qualitatively different sets of questions as shown in

	mean	std	# examples
QABriefs (Fan et al., 2020)	2.88	1.20	210
Ours	3.60	1.19	210
p-value	≤ 0.0001		
mean diff	0.72		
95% CI	0.48 - 0.97		

Table 6.5: Results from user study on helpfulness (rated 1-5) of a set of generated subquestions for claim verification. We conduct a t-test over the collected scores.

	Macro-F1	Micro-F1	MAE
Question aggregation	0.30	0.29	1.05
Question aggregation*	0.46	0.45	0.73
Random (label dist)	0.16	0.18	1.68
Most frequent	0.06	0.23	1.31

Table 6.6: Claim classification performance of our question aggregation baseline vs. several baselines on the development set. MAE denotes mean absolute error.

Figure 6.5. We notice: (1) Their questions are less comprehensive, often missing important aspects of the claim. (2) Their questions are broader and less focused on the claim. We instructed annotators to provide the **source** of the annotated subquestions from either claim or verification document. For example, questions like "*What are Payday lenders?*" in the figure will not appear in our dataset as the justification paragraph does not address such question. (Fan et al., 2020) dissuaded annotators from providing binary questions; instead, they gather answers to their subquestions after the questions are collected. We focus on binary questions whose verification could help verification of the full claim. See the original paper (Chen et al., 2022) for more examples of the comparison.

User Study To better quantify the difference, we also conduct a user study in which we ask an annotator to rate how useful a set of questions (without answers) are to determine the veracity of a claim. On 42 claims annotated by both approaches,

annotators score sets of subquestions on a Likert scale from 1 to 5, where 1 denotes that knowing the answers to the questions does not help at all and 5 denotes that they can accurately judge the claim once they know the answer. We recruit annotators from MTurk. We collect 5-way annotation for each example and conduct the t-test over the results. The details can be found in the original paper (Chen et al., 2022).

Table 6.5 reports the user study results. Our questions achieve a significantly higher relevance score compared to questions from QABriefs. This indicates that we can potentially derive the veracity of the claim from our decomposed questions since they are binary and highly relevant to the claim.

6.5.3 Deriving the Veracity of Claims from Decomposed Questions

Is the veracity of a claim sum of its parts? We estimate whether answers to subquestions can be used to determine the veracity of the claim.

We predict a veracity score $\hat{v} = \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}[a_i = 1]$ equal to the fraction of subquestions with yes answers. We can map this to the discrete 6-label scale by associating the labels *pants on fire*, *false*, *barely true*, *half true*, *mostly true*, and *true* with the intervals $[0, \frac{1}{6}), [\frac{1}{6}, \frac{2}{6}), [\frac{2}{6}, \frac{3}{6}), [\frac{3}{6}, \frac{4}{6}), [\frac{4}{6}, \frac{5}{6}), [\frac{5}{6}, 1]$, respectively. We call this method **question aggregation**. We use the 50 claims and the corresponding questions from the **Validation-sub** set for evaluation. We also establish the upper bound (**question aggregation***) for this heuristic by having one of the authors remove unrelated questions. On average, 0.3 questions are removed per claim.

Table 6.6 compares our heuristics with simple baselines (random assignment and most frequent class assignment). Our heuristic easily outperforms the baselines, with the predicted label on average is only shifted by one label, e.g., *mostly true* vs. *true*. This demonstrates the potential of building a more complex model to

aggregate subquestion-answer sets.

Our simple aggregation suffers in the following cases: (1) The subquestions are not equal in importance. The first example in Figure 6.4 contains two yes subquestions and two no subquestions, and our aggregation yields *half-true* label, differing from gold label *barely-true*. (2) Not all questions are relevant. As indicated by **question aggregation***, we are able to achieve better performance after removing unrelated questions. (3) In few cases, the answer to a question could inversely correlate with the veracity of a claim. For example, the claim states "Person X implied Y" and the question asks "Did person X not imply Y?" We think all of the cases can be potentially fixed by stronger models. For example, a question salience model can mitigate (1) and (2), and promotes researches about understanding core arguments of a complex claim. We leave this as future work.

6.6 Evidence Retrieval with Decomposition

Lastly, we explore using claim decomposition for retrieving evidence paragraphs to verify claims. Retrieval from the web to check claims is an extremely hard problem (Singh et al., 2021a). We instead explore a simplified proof-of-concept setting: retrieving relevant paragraphs from the full justification document. These articles are lengthy, containing an average of 12 paragraphs, and with distractors due to entity and concept overlap with the claims.

We aim to show two advantages of using the decomposed questions: (1) The implied questions contain information helpful to retrieve evidence beyond the lexical information of the claim. (2) We can convert the subquestions to statements and treat them as hypotheses to apply the off-the-shelf NLI models to retrieve evidence that entails such hypotheses (Chen et al., 2021).

	per subquestion	per example (claim)
avg # of paras	12.4	12.4
% of context	87.6	68.8
% of support	5.4	12.0
% of refute	8.0	19.2
Fleiss Kappa	0.42	0.42

Table 6.7: Evidence paragraph retrieval data statistics on Validation-sub dataset (50 claims).

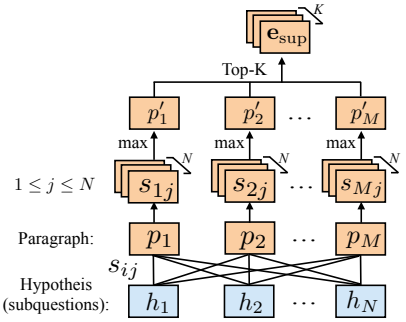


Figure 6.6: Illustration of evidence paragraph retrieval process. The notations corresponds to our descriptions in Section 6.6. K is a hyperparameter controlling the number of passages to retrieve.

Evidence Paragraph Collection We first collect human annotation to identify relevant evidence paragraphs. Given the full PolitiFact verification article consisting of m paragraphs $\mathbf{p} = (p_1, \ldots, p_m)$ and a subquestion, annotators find paragraphs relevant to the subquestion. As this requires careful document-level reading, we hire three undergraduate linguistics students as annotators. We use the 50 claims from the Validation-sub set and present the annotators with the subquestions and the articles. For each subquestion, for each paragraph in the article, we ask the annotators to choose whether it served as context to the subquestion or whether it supports/refutes the subquestion. The statistics and inter-annotator agreement is shown in Table 6.7. Out of 12.4 paragraphs on average, 3-4 paragraphs were directly relevant to the claim and the rest of paragraphs mostly provide context.

Model	Decomposed claim		Original
	predicted	gold	claim
MNLI	41.0	48.8	35.2
NQ-NLI	38.8	34.5	40.9
DocNLI	**44.7**	**59.6**	36.9
BM25	36.2	47.5	39.2

Table 6.8: Evidence retrieval performance (F1 score) with the decomposed claims (from predicted and annotated (gold) subquestions) and the original claim on the Validation-sub set. A random baseline achieves 24.9 F1 and human annotators achieve 69.0 F1.

Experimental Setup We experiment with three off-the-shelf RoBERTa-based (Liu et al., 2019) NLI models trained on three different datasets: MNLI (Williams et al., 2018a), NQ-NLI (Chen et al., 2021), and DocNLI (Yin et al., 2021). We compare the performance of NLI models with random, BM25, and human baselines.

We first convert the corresponding subquestions $\mathbf{q} = q_1, ..., q_N$ of claim c to a set of statements $\mathbf{h} = h_1, ..., h_n$ using GPT-3 (Brown et al., 2020).[5] We find that with only 10 examples as demonstration, GPT-3 can perform the conversion quite well (with an error rate less than 5%). For more information about the prompt see the original paper (Chen et al., 2022) for details.

To retrieve the evidence that **supports** the statements, we treat the statements as hypotheses and the paragraphs in the article as premises. We feed them into an NLI model to compute the score associated with the "entailment" class for every premise and hypothesis pair. Here, the score for paragraph p_i and hypothesis h_j is defined as the output probability $s_{ij} = P(\text{Entailment} \mid p_i, h_j)$. We then select as evidence the top k paragraphs by score across all subquestions: for paragraph p_i, we define $p'_i = \max(\{s_{ij} \mid 1 \leq j \leq N\})$, which denotes for each hypothesis

[5] We release the automatically converted statements and the negations for all of the subquestions in the published dataset.

from 1 to N that the jth hypothesis h_j achieves the highest score with p_i. Then $\mathbf{e_{sup}} = \{p_i \mid i \in \text{Top-K}(\{p'_1, ..., p'_M\})\}$. We set k to be the number of the paragraphs that are annotated with either support or refute. Figure 6.6 describes this approach.

To retrieve the evidence that **refutes** the statements, we follow the same process, but with the negated hypotheses set **h** generated by GPT3. (Note that our NLI models trained on NQ-NLI and DocNLI only have two classes, entailed and not entailed, and not entailed is not a sufficient basis for retrieval.) The final evidence set is obtained by merging the evidence from the *support* and *refute* set. This is achieved by removing duplicates then taking Top-K paragraphs according to the scores.

BM25 baseline model uses retrieval score instead of NLI score. **The random baseline** randomly assign support, refute, neutral labels to paragraphs based on the paragraph label distribution in Table 6.7. **Human performance** is computed by selecting one of the three annotators and comparing their annotations with the other two (we randomly pick one annotator if they do not agree), taking the average over all three annotators. This is not directly comparable to the annotations for the other techniques as the gold labels are slightly different.

Results The results are shown in Table 6.8. We see that **the decomposed questions are effective to retrieve the evidence.** By aggregating evidence from the subquestions, both BM25 and the NLI models can do better than using the claim alone, except for the case of using DocNLI, and BM25 with the predicted decomposition. The best model with gold annotations (59.6) is close to human performance (69.0) in this limited setting, indicating that the detailed and implied information in decomposed questions can help gathering evidence beyond the surface level of the claim.

DocNLI outperforms BM25 on both the annotated decomposition and the predicted decomposition. This demonstrates the potential of using the NLI models to aid the evidence retrieval in the wild, although they must be combined with decomposition to yield good results.

6.7 Related Work

Fact-checking Vlachos and Riedel (2014b) proposed to decompose the fact-checking process into three components: identifying check-worthy claims, retrieving evidence, and producing verdicts. Various datasets have been proposed, including human-generated claims based on Wikepedia (Thorne et al., 2018; Chen et al., 2019b; Jiang et al., 2020; Schuster et al., 2021; Aly et al., 2021b), real-world political claims (Wang, 2017; Alhindi et al., 2018; Augenstein et al., 2019; Ostrowski et al., 2021; Gupta and Srikumar, 2021), and science claims (Wadden et al., 2020; Saakyan et al., 2021). Our dataset focuses on real-world political claims, particularly more complex claims than past work which necessitate the use of decompositions.

Our implied subquestions go beyond what is mentioned in the claim, asking the intention and political agenda of the speaker. Gabriel et al. (2022) study such implications by gathering expected readers' reactions and writers' intentions towards news headlines, including fake news headlines.

To produce verdicts of the claims, other work generates explanations for models' predictions. Popat et al. (2017, 2018); Shu et al. (2019); Yang et al. (2019a); Lu and Li (2020) presented attention-based explanations; Gad-Elrab et al. (2019); Ahmadi et al. (2019) used logic-based systems, and Atanasova et al. (2020); Kotonya and Toni (2020) modeled the explanation generation as a summarization task. Combining answers to the decomposed questions in our work can form an explicit explanation of the answer.

Question Generation Our work also relates to question generation (QG) (Du et al., 2017), which has been applied to augment data for QA models (Duan et al., 2017; Sachan and Xing, 2018; Alberti et al., 2019a), evaluate factual consistency of summaries (Wang et al., 2020a; Durmus et al., 2020; Kamoi et al., 2022), identify semantic relations (He et al., 2015; Klein et al., 2020; Pyatkin et al., 2020), and identify useful missing information in a given context (clarification) (Rao and Daumé III, 2018; Shwartz et al., 2020; Majumder et al., 2021). Our work is most similar to QABriefs (Fan et al., 2020), but differs from theirs in two ways: (1) We generate yes-no questions directly related to checking the veracity of the claim. (2) Our questions are more comprehensive and relevant to the claim.

6.8 Chapter Summary

We present a dataset containing more than 1,000 real-world complex political claims with their decompositions in question form. With the decompositions, we are able to check the explicit and implicit arguments made in the claims. We also show the decompositions can play an important role in both evidence retrieval and veracity composition of an explainable fact-checking system. We believe that this dataset can further promote building explainable fact-checking systems and analyzing complex claims.

Chapter 7

Fact Verification with Evidence Retrieved in the Wild

This chapter is based on Chen et al. (2023).[1]

7.1 Introduction

To combat the rise of misinformation, the NLP community studied providing automated tools to assist with fact-checking. In Chapter 6, we used human-written fact-checking articles as a retrieval corpus to retrieve evidence, which is not useful to build a real fact-checking system. Additionally, many prior work studies fact-checking have focused on crowd-sourced claims (Thorne et al., 2018; Jiang et al., 2020; Schuster et al., 2021; Aly et al., 2021a), which do not accurately represent the complexities of actual claims that fact-checkers deal with. Other work that does tackle real-world claims either relies on access to a document set that contains the "gold" evidence (Ferreira and Vlachos, 2016; Alhindi et al., 2018; Hanselowski et al., 2019; Atanasova et al., 2020) or conducts unconstrained retrieval (Augenstein et al., 2019), which may retrieve articles written by fact-checkers. This assumption fails to account for the challenges in retrieving the raw evidence in the wild.

We simulate realistic fact-checking scenarios, handle complex political claims, and study a retrieval setting that aligns with the fact-checker's workflow. We re-

[1] Jifan Chen, Grace Kim, Aniruddh Sriram, Greg Durrett, and Eunsol Choi. 2023. Complex Claim Verification with Evidence Retrieved in the Wild. arXiv preprint arXiv:2305.11859.
Jifan Chen initialized the research project, conducted experiments, analyzed data and wrote the paper.

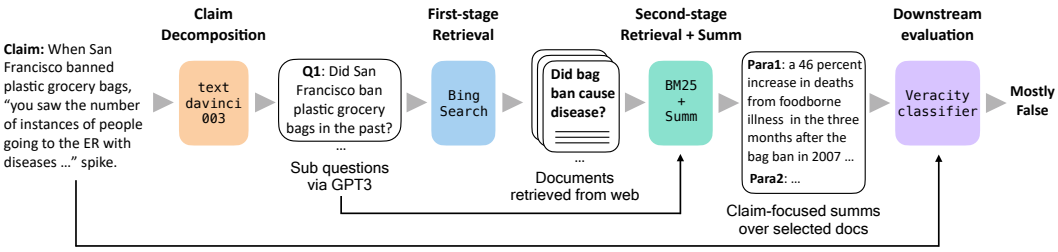


Figure 7.1: Overview of our pipeline: a claim is first decomposed into several yes/no questions (section 7.3.1), then we pipe the questions through two stages of retrieval (section 7.3.2 and section 7.3.3) to select the most relevant paragraphs. Finally, we generate a claim-focused summarization (section 7.3.4) and feed it to a veracity classifier to get the final veracity label (section 7.3.5).

trieves evidences from the Web, restricted to document authored before the time of the claim and not documents sourced from fact-checking websites themselves. To handle this challenging setting, we propose a pipeline that builds upon the strength of large-scale language models and findings from prior studies. Following the approach of Chapter 6, we first decompose a claim into a series of sub-questions, targeting both explicit and implicit aspects of the claim. Each sub-question is fed into a commercial search engine to retrieve relevant documents, with restrictions described above to avoid "cheating." Then, we perform a second stage of fine-grained retrieval to isolate the most relevant portions of the documents.

Then, identifying relevant information from lengthy documents becomes a key challenge: information from prior to a claim is often only obliquely related to that claim and needs significant reshaping or processing. To achieve this, we employ state-of-the-art language models (Brown et al., 2020; Ouyang et al., 2022) to generate claim-focused summaries. These summaries can both serve as explanations for users of the system as well as input to a classifier to determine the final veracity based on these summaries.

Evaluating individual components of our pipeline is challenging due to the

absence of gold annotations at each stage. Instead, we evaluate the downstream veracity classification performance. Since the veracity judgement is inherently subjective (Lim, 2018) the classification performance alone may not fully represent the system's effectiveness. Therefore, we conduct a human study to evaluate the comprehensiveness and faithfulness of the claim-focused summaries generated in the final step of our pipeline.

We apply our pipeline to ClaimDecomp proposed in Chapter 6, a dataset containing 1,200 real-world complex political claims with veracity labels from professional fact-checkers. Performance on veracity classification show that: (1) our constrained retrieval setting is indeed much harder than "unrestricted" retrieval settings; (2) introducing evidence retrieved from the web leads to performance gains compared to automatic fact-checking without evidence, though there remains a significant gap between an oracle classifier performance based on human-written justifications; (3) the sub-questions are crucial for obtaining high-quality raw documents from the web compared to using the original claim alone.

Our human study further indicates that: (1) claim-focused summaries generated through are faithful most of the time and are helpful for both machine and humans to fact-check a claim; (2) the retrieved evidence is often relevant to some aspects of the claim, but can rarely cover all of its aspects, suggesting that finding the correct raw evidence in the wild is the core challenge in building automatic fact-checking systems. We hope our work will spark NLP research in assisting fact-checkers in realistic scenarios.

7.2 Background

Many of the widely used fact verification benchmarks, such as FEVER (Thorne et al., 2018), HoVer (Jiang et al., 2020), and VitaminC (Schuster et al., 2021), fo-

cus on crowd-sourced claims derived from Wikipedia. For example, claims in the FEVER dataset typically require checking a single aspect like "*Oliver Reed was a film actor.*" These claims are checkable by retrieving evidence from Wikipedia and can be annotated by crowdworkers at scale, but they do not reflect the complexities of real-world political claims.

Earlier studies (Vlachos and Riedel, 2014b; Wang, 2017; Pérez-Rosas et al., 2018) on fact-checking political claims typically considered claim alone as an input to an automated system. By not having evidence, the judgment about the claim is necessarily based on surface-level linguistic patterns and cannot account for subtle factual errors. Research that does incorporate evidence either assumes access to justifications provided by fact-checkers (Vlachos and Riedel, 2014b; Alhindi et al., 2018; Hanselowski et al., 2019; Atanasova et al., 2020) or evidences from *unconstrained* retrieval (Popat et al., 2017, 2018; Augenstein et al., 2019), which frequently yields evidence sets containing information from fact-checking websites themselves. Fan et al. (2020) explore generating questions to assist humans in retrieving evidence from the web, but they only evaluate their system with a human in the loop, who can aggressively filter irrelevant retrieval results.

To our knowledge, we present first automatic fact-checking with a realistic retrieval pipeline using evidence that would be available to fact-checkers at the time a claim was made. As a result, this setting is very challenging and many claims are not checkable. We therefore emphasize the evidence that our system returns as a way of assisting human fact-checkers; we believe this realistic task setting and our corresponding evaluation should be reused in future work.

Need for these tools Our work further shifts the focus away from the evaluation on classification accuracy alone. Accuracy on truth labels provided by PolitiFact is a

proxy metric we use to evaluate our systems. However, fact-checking experts argue that the task is too subjective and complex to be automatable in the near term (Graves, 2018; Nakov et al., 2021). Part of this arises from the fact that information needed to check claims is not always available on the public Internet (Singh et al., 2021b). Closed-book systems like GPT-4 cannot access this information at all, and even systems like WebGPT (Nakano et al., 2022) struggle to access all the requisite information. Returning information on a best-effort basis and providing detailed evidence to enable a human to assist in the judgment can help overcome issues with returning judgments from error-prone AI systems (Bansal et al., 2021).

7.3 Methodology

Our pipeline (Figure 7.1) consists of five parts, namely claim decomposition, raw document retrieval, fine-grained retrieval, claim-focused summarization, and veracity classification. We describe each part below.

7.3.1 Subquestion Decomposition

Given a real-world complex claim, we first decompose it into a set of yes/no questions of which the answers are essential to fact-check the claim, as it has been shown in Chapter 6 that the decomposition is both helpful to retrieve relevant evidence and make a judgment about the final veracity. We pick four input-decomposition pairs from the human annotations of Chapter 6 to form a few-shot prompt and feed it to OpenAI's `text-davinci-003` which is a GPT3 (Brown et al., 2020) model from the Instruct series (Ouyang et al., 2022), to generate the sub-questions. As fact-checking often requires checking multiple aspects of a claim, we generate a set of unique questions by multiple rounds of sampling until we generate 10 different questions. An example of the decomposition is shown in Figure 7.2.

7.3.2 First-stage Retrieval

For each question generated in the previous step, we feed it to a commercial search engine API to collect the top 20 documents. We filter the documents from fact-checking websites, after which we pick the top 10 documents from the remaining documents. In this chapter, we use the Bing Search API.[2]

Date Indexing To simulate realistic fact-checking scenarios, that is, an automatic fact-checking system should be able to verify a claim as soon as it has been made. This indicates that the system can only access documents that are available before the claim is made. Additionally, we investigate the extent to which the evidence leaks if we conduct free retrieval without timestamps. Therefore, we conduct two rounds of web retrieval with and without the timestamp of a claim.

Content Extraction We extract the actual content from the page URLs retrieved by the Bing Search API using two tools: `html2text`[3] and `readability-lxml`.[4] It should be noted that some URLs are protected[5] and cannot be scraped. The statistics of the retrieved documents can be found in Table 7.1. We combine the two sets of documents as the corpus for our experiments.

7.3.3 Second-stage Retrieval

Most of the documents collected from the previous step contain only small snippets relevant to the claim, if they are relevant at all. Thus, we conduct a second-stage retrieval to pick the most relevant text spans regarding the claim. Specifically,

[2] /www.microsoft.com/en-us/bing/apis/bing-web-search-api
[3] https://github.com/Alir3z4/html2text/
[4] https://github.com/buriy/python-readability
[5] Paywall, PDFs, and anti-scraping measures.

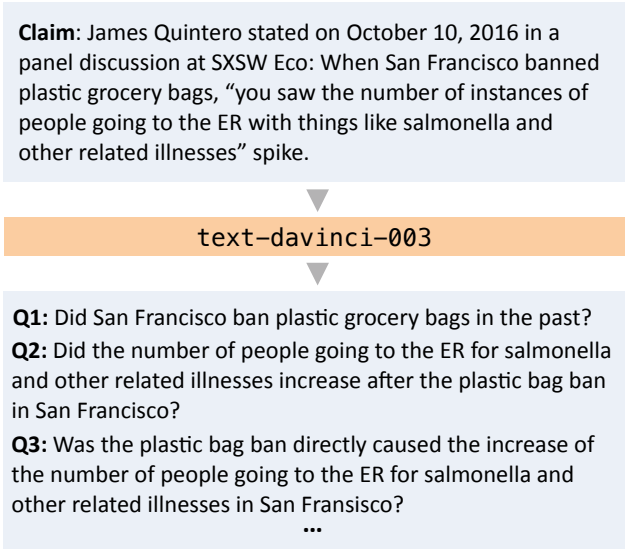


Figure 7.2: A demonstration of our claim decomposition process. We decompose each claim into 10 unique questions. We only show three questions for simplicity.

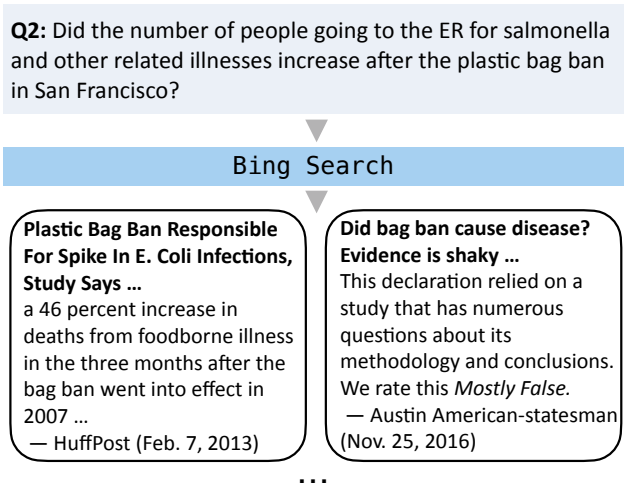


Figure 7.3: Two documents returned by searching Q2 (generated in the previous stage) through the search engine. Here we see the right page is created one month after the claim and it cites the article written by PolitiFact, which leaks core information thus problematic to use as raw evidence.

	# retrieved	# scraped	# words
w/ timestamp	66.7	45.0	1,561
w/o timestamp	70.4	47.8	1,660
combined	122.5	82.8	1,623

Table 7.1: The statistics for the retrieved documents obtained through the first-stage retrieval, averaged over all instances. Approximately one-third of the documents are protected and cannot be scraped. Furthermore, there is not much overlap between the two separate retrievals.

we segment the documents into text spans containing k words with a stride of $\frac{1}{2}k$ words. In our experiments, k is set to 30. Following Chapter 6, we employ `BM-25` to retrieve the top-10 highest-scored text spans, expanding these spans with a $\pm$ 150-word context. If two text spans overlap, they are merged to form a larger span. This process typically yields large text segments from more than five distinct documents.

7.3.4 Claim-Focused Summarization

Since the documents retrieved in the previous step can contain several thousand words, it becomes impractical for both humans and models to make a judgment based on such extensive content. Consequently, we utilize state-of-the-art LMs, specifically `text-davinci-003`, to summarize each retrieved document *separately* with respect to the claim.

Enforcing summary instead of judgment During a pilot study, we discovered that even when explicitly instructing the model to generate only a summary, `text-davinci-003` tends to make unfaithful judgments about the document by producing a verdict such as "therefore, the claim is refuted by the document," even when given irrelevant documents.

To circumvent this issue, we investigate two types of prompt engineering. For the zero-shot prompt, we explicitly instruct the model not to make any judgments about the stance of the given document. For the few-shot prompt, we select four documents and manually write summaries. For documents that are not relevant to the claim, we explicitly write "the document is not relevant to checking the claim" in the prompt. In section 7.5.1, we show that the few-shot prompting makes the generated summaries more faithful. We also compare against a variant of our system using a single summary, rather than one per document. We use zero-shot prompting for this **single-zero** variant.

7.3.5 Veracity Classification

The final stage of our pipeline involves making a judgment based on the summaries generated in the previous stage. We train a DeBERTa-large (He et al., 2020) model to perform 6-way classification. The input to the DeBERTa model is a concatenation of the claim and the summaries of the retrieved documents, while the output is one of the six labels from Chapter 6 (true, mostly true, half true, barely true, false, and pants-on-fire).

7.4 Automatic Claim Verification Evaluation

Our main automatic evaluation is on claim veracity prediction (Wang, 2017), evaluating our entire pipeline.

7.4.1 Experimental Settings

Data We use the data from Chapter 6 which contains 1,200 complex claims from PolitiFact. Each claim is labeled with one of the six veracity labels, a justification paragraph written by expert fact-checkers, and sub-questions annotated by prior

Temporal	Site	Acc	Soft Acc	Ma-F1	MAE
All	All	63.5	76.5	62.2	0.49
All	Non-FC	40.0	54.5	39.8	0.88
Before	All	40.0	57.5	39.8	0.84
Before	Non-FC	39.0	52.0	39.6	0.89
Claim only		27.0	36.0	16.1	1.1
Claim + Just (oracle)		51.5	68.0	46.3	0.61

Table 7.2: Final veracity classification performance given different retrieval constraints. We report the test set performance by choosing the best model over 5 runs using different random seeds on the development set. The top block are our full system with constraints over what is retrieved. Red indicates using oracle information.

work.

Evaluation Metric Following Chapter 6, we report several evaluation metrics, including accuracy (Acc), mean absolute error (MAE), Macro-F1, and soft accuracy (soft Acc). The soft accuracy is calculated by merging True and Mostly True as one label, and Pants on Fire, False, and Mostly False as another label, while Half True remains as a separate label.

Since the training set is small, we train the classifier five times with different random seeds and report the test set performance using the model that achieves the best performance on the development set over the 5 runs.

7.4.2 Comparison Systems

The default version of our system for our experiments is to use `text-davinci-003` generated subquestions to do both site-constrained and time-constrained web retrieval. Then use those sub-questions to do the second-stage retrieval and use `zero-shot-003` as the summarization model.

Variants of our systems based on different constraints. As discussed in section 7.3.2, we constrain document access based on its timestamps and whether it comes from fact-checking websites. We experiment with four distinct variants by combining the two constraints.

Claim-only We concatenate the metadata, including the speaker and the venue of the claim, with the claim itself, and feed the resulting text into the classifier. This approach serves as a lower bound for the veracity classification. This follows the setting used by Wang (2017).

Claim+Just We extend the *Claim-only* baseline by appending the human-written justification paragraph[6] to the claim. Note this is the oracle setting and sets the upper bound for the veracity classification.

7.4.3 Comparison: Constrained vs. Unconstrained Search

We first situate our work with respect to baselines and past systems by varying the retrieval condition. Specifically, we experiment with both **temporal** constraints and **site** constraints: we can allow all pages or just pages occurring **before** the date of the claim, and we can additionally constrain the valid sites to be **non-fact-checking (non-FC)** sites. The unconstrained "All/All" setting used in MultiFC (Augenstein et al., 2019)

Table 7.2 reports the performance of our system with baselines. We note first that **adding temporal or site constraints dramatically reduces the performance**. This demonstrates that unconstrained retrieval over the web works largely because it is able to retrieve fact-checks that were published after the claim

[6]We remove the sentence containing the label in the justification.

	Evidence Distillation			Performance			
	FSR	SSR	Summary	Acc	Soft Acc	Macro-F1	MAE
		Claim only		27.0	36.0	16.1	1.1
		Claim + Justification (oracle)		51.5	68.0	46.3	0.61
			Our Best System				
Ⓑ	subQs	subQs	multi-zero	39.0	52.0	39.6	0.89
			Ablation on first-stage retrieval				
①	Claim			36.5	50.0	36.9	1.03
			Ablation on second-stage retrieval				
②		Claim		39.5	53.5	30.8	0.91
③		Gold subQs		41.0	51.0	41.2	0.95
④		Justification		40.0	53.5	40.0	0.90
			Ablation on summarization				
⑤			multi-few	37.5	53.0	37.1	0.94
⑥			single-zero	42.0	53.0	40.9	0.85
⑦			no summary (raw doc)	37.5	51.0	31.9	0.99

Table 7.3: End-to-end factchecking performance. We ablate various stages of the model (FSR: first-stage retrieval; SSR: second-stage retrieval). Retrieval using sub-questions is quite helpful at the first stage, but less so at the second stage. Using our GPT-3 summarization is important (compare to Raw Docs). Red indicates using oracle information.

was released, which greatly simplify the problem by synthesizing the raw evidence before it is fed to our system. In fact, this performance is even higher than the claim + justification setting, which is already an oracle because it shows an explanation of the gold fact-checking decision.

Moreover, when adding time constraints, we see a further drop in performance, as this eliminates the cases where other sites might reference fact-checking pages, as illustrated in Figure 7.3. We suggest follow-up work on retrieval focuses on those constrained settings.

Finally, if we compare the performance of *claim-only* and other models that

use retrieval, we see a notable improvement over all four of our metrics, showing that **retrieving and summarizing evidence is helpful to predict the veracity label, even in the constrained setting**. We will now investigate more deeply which parts of our pipeline are responsible for this performance.

7.4.4 Ablations

We ablate the first-stage retrieval, second-stage retrieval, the summarization model, and the classifier to understand how each individual component contributes to the final performance. The results are shown in Table 7.3.

Referring back to Figure 7.1, we modify the following steps of the pipeline

- **First-stage retrieval** Rather than retrieving with sub-questions (**subQs**), we instead perform our search with the raw claim (**Claim**).
- **Second-stage retrieval** Rather than retrieving with sub-questions (**subQs**), we instead perform our search with the raw **Claim**, **Gold subQs** from Chapter 6, or **Justification**, which uses oracle information.
- **Summ**: **multi-zero** or **multi-few** differ in the prompt; **single-zero** produces a single summary of all the retrieved documents. **no summary** means directly using the unsummarized documents as input to classifier.

We have the following observations:

The decomposed subquestions are effective for retrieving relevant documents from the web. Comparing Ⓑ and ①, by changing the input of first-stage retrieval to the original claim instead of sub-questions, we observe a notable decrease in classification performance across all evaluation metrics. This can be attributed to

the fact that the generated sub-questions encompass multiple aspects of the claim, enabling the search engine to locate relevant information more easily.

Once we have the raw documents from the web, using either questions or claims does not make a difference in the second-stage retrieval. Systems ②, ③, and ④ yield only slight differences in classifier performance, even in spite of the fact that ④ uses the human-written justification for second-stage retrieval. We believe this is because we expand the retrieved text span by a ±150 words of context window and feed the context through a summarization model. As a result, this retrieval step does not need to be all that precise.

Claim-focused summarization is essential for achieving optimal performance. System ⑦ shows worse performance than Ⓑ across all metrics, suggesting that some summarization step is important. This may result from two primary factors: (1) The document length exceeds the context window capacity of DeBERTa, causing crucial information to be truncated. (2) DeBERTa is not sufficiently robust to discern the most relevant information given an excessive amount of context. Differences in the summarization strategy (⑤ and ⑥) did not yield major changes here but did have an impact on our human evaluation in the next section.

7.5 Human Study of the Claim-focused Summaries

As discussed in section 7.4.4, incorporating the claim-focused summarization generated by GPT3 substantially improves the performance of the final-stage classifier. Nevertheless, it is widely recognized that large language models sometimes generate untruthful content (Bommasani et al., 2021; Chowdhery et al., 2022; Ouyang et al., 2022), which could introduce bias in the final-stage classifier's decision. Fur-

thermore, as pointed out by Lim (2018), the accuracy of veracity classification alone does not entirely reflect the system's overall effectiveness, as certain labels such as "false" and "barely-true" may be ambiguous. We believe the true measure of our system's utility lies in the full package of summarized evidence it returns rather than just the label. Therefore, in this section, we carry out two human studies, namely, comprehensiveness and faithfulness, to better understand the system's behavior.

7.5.1 Faithfulness Evaluation

We assess the frequency and degree to which the language model generates untruthful content during query focused summarization. For each document and summary pair, annotators choose one of four labels below:

- **Faithful:** the summary accurately represents the meaning and details of the original document.
- **Minor Factual Error:** some details are not aligned with the original document, but the overall message remains intact.
- **Major Factual Error:** there are factual errors that result in the summary misrepresenting the original document.
- **Completely Wrong:** the language model hallucinates content that completely alters the meaning of the original document.

Besides selecting a label, we ask annotators to provide a natural language justification for their choices.

We randomly pick 50 claims which contain 200 document-summary pairs from the development set of ClaimDecomp. We mainly compare the two types of summaries we discussed in section 7.3.4, namely, `zero-shot-003` and `few-shot-003`

Summ-type	F	Minor	Major	NF	Avg score
`zero-shot-001`	65.8%	9.2%	20%	5%	3.45
`zero-shot-003`	66%	18%	16%	0%	3.50
`few-shot-003`	82.5%	6.5%	8.5%	2.5%	3.69

Table 7.4: Human evaluation results on the same 200 document-summary pairs from 50 claims we randomly picked from zero-shot and few-shot summaries based on `text-davinci-003`. "F" denotes the summary is factual and "NF" denotes the summary is completely wrong. Few-shot prompting helps the model make fewer factual errors.

(both in the "multi-" setting, one summary per document). We also compare the summaries generated through `text-davinci-001` to see how the faithfulness varies for different models. We recruited annotators from Amazon Mechanical Turk with a qualification test, which selects workers that get more than three out of five examples correct and provide reasonable explanations for their choices. Total of 17 workers participated in a 3-way annotation.

The annotations agree with a Fleiss Kappa score of 0.30. Although the agreement is not high, we check the justifications given by the annotators and find that many of the disagreements are because of subjectivity regarding a factual error.

We compute a consensus annotation via majority vote. We also assign numerical scores to each label, where "Faithful", "Minor", "Major", and "Completely Wrong" correspond to 4, 3, 2, and 1 respectively. We compute the average score for zero-shot and few-shot summaries according to the aggregated labels[7].

Results The results are shown in Table 7.4. We see that **few-shot prompting substantially decreases the chance of hallucinations in the summaries**.

[7]If all annotators disagree, we compute the average score and pick the nearest label

When combining "Factual" and "Minor", we see 89% of the summaries are good enough to be used as evidence for the classifier. Comparing the performance of `zero-shot-001` and `zero-shot-003`, we find that the weaker model makes more major factual errors. These two observations indicate that with stronger models and better prompts, we can expect these summarization models to improve further.

Figure 7.4 shows 3 examples containing unfaithful content. We see that the "Minor" error does not affect the interpretation of the original document while "Major" and "Completely Wrong" alters the view.

7.5.2 Comprehensiveness Evaluation

The goal of the comprehensiveness study is to measure the extent to which the claim-focused summaries are able to address the claim. However, assessing the extent to which a claim is verified is nontrivial. To do this, we leverage the human-annotated yes/no sub-questions presented in Chapter 6 as a proxy for evaluating the quality of our summaries. Specifically, for each sub-question, we ask annotators to determine whether the question is answerable or not. Sometimes the questions cannot be directly answered but can be inferred from the content of the summaries, or the summary at least contains relevant information. In such cases, we ask annotators to choose "partially answerable". If the question is deemed answerable, we ask the annotator to indicate whether the answer is "Yes" or "No" based on their best judgment. In addition to the answer, we also ask annotators to give a justification for their answers.

We recruit annotators from Mechanical Turk following the same protocol of the faithfulness study. We select 15 workers and conduct a 3-way annotation over 161 questions from the same 50 claims we picked for the human study of faithfulness. The annotation agrees with a Fleiss Kappa score of 0.32.

Summ-type	Ans	Partially Ans	UnAns
`zero-shot-003`	47.8%	22.4%	29.8%
`few-shot-003`	42.9%	21.1%	36.0%

Table 7.5: Human evaluation results on 161 sub-questions from the same 50 claims we picked for the human study on faithfulness. "Ans", "P-Ans", and "UnAns" denotes the number of questions that is answerable, partially answerable, and unanswerable respectively.

	Faithful	Minor	Major+ Wrong	Total
Answerable	4	2	0	6
Partially Ans	6	1	1	8
Unanswerable	18	7	12	36
Total	28	10	12	50

Table 7.6: **Claim-level** statistics of `few-shot-003` by taking faithfulness and comprehensiveness into consideration. The claim-level labels are derived from the subparts as defined in section 7.5.3.

Results The results are presented in Table 7.5. We see that zero-shot summaries yield more answerable questions than few-shot summaries. However, based on the faithfulness results, we believe this may be because hallucinations in zero-shot summaries make the questions answerable in an unfaithful way: the system imputes information that seems to help, but which is not supported by the document.

Nevertheless, the few-shot summaries allow us to partially address over 60% of the **gold annotated** sub-questions in the PolitiFact justification. We find this result encouraging: it indicates that even though the system does not have access to these (often subtle) factors, it can return relevant information to enable a human annotator to make a judgment about them.

Claim: Facebook posts stated on September 29, 2021: The White House “created a fake set for (President Joe) Biden to get his booster shot. The entire Biden presidency is one giant charade.”
Document Title: President Joe Biden Gets COVID Booster Vaccine Shot - PEOPLE.com
Content: President Joe Biden received his COVID-19 booster shot .. Biden received his first dose of Pfizer-BioNTech novel coronavirus vaccine in December at Delaware 's ChristianaCare Hospital. His second shot was administered there three weeks later. Both shots were televised ...
Summary: On Monday, President Joe Biden received his COVID-19 booster shot on camera, administered by a nurse with the White House Medical Unit. Biden has previously received both of his initial doses of the Pfizer-BioNTech novel coronavirus vaccine in December at Delaware's ChristianaCare Hospital.
Minor Error: The summaries says both shots are received in December while the document says the first one was in December and the second one was three weeks later. The summary slightly changes the information in the original document.

Claim: Elizabeth Warren stated on November 20, 2019 in a debate: "Today in America — a new study came out — 20 years out, whites who borrowed money, 94% of them have paid off their student loan debt; 5% of African Americans have paid it off.”
Document Title: More Evidence Student Borrowers Prepay Their Loans - Forbes
Content: Most borrowers will still fully pay off their loans before they are required to do so ... Borrowers pay off almost all loans (94 %) with a final payment larger than the scheduled monthly payments they are required to make ...
Summary: A new report from the Consumer Financial Protection Bureau found that 94% of borrowers pay off their student loans before the scheduled payoff date, typically doing so in five years or less ...
Major Error: The document says borrowers pay off 94% of the loan not 94% of borrowers pay off their loans

Claim: Andrew Giuliani stated on May 18, 2021 in a news conference: "The one good thing about the antibodies if you’ve had it, is it actually is even better than the vaccine, and here's why. With the vaccine you can still transmit, with the antibodies you can't transmit.”
Document Title: COVID-19: Long-term effects - Mayo Clinic
Content: It involves extreme fatigue that worsens with physical or mental activity , but doesn't improve with rest ... What should you do if you have post-COVID-19 syndrome symptoms ? If you 're having symptoms of post-COVID-19 syndrome , talk to your health care provider ...
Summary: The Centers for Disease Control and Prevention states that there is no evidence to suggest that people who have recovered from COVID-19 and have antibodies are not able to transmit the virus.
Completely Wrong: The document is about the long-term effects of COVID-19. However, model is likely utilizing its parameterized knowledge and draws the conclusion directly.

Figure 7.4: Three examples from faithful evaluation (Section 7.5.1), showing the cases of minor error, major error, and completely wrong respectively. Red marks denote the mismatches between the summary and the document.

7.5.3 Holistic Evaluation

In the previous sections, the comprehensiveness and faithfulness studies are conducted on the sub-question level and the document level respectively. We investigate the **claim-level** statistics in this section. We aim to answer in a holistic fashion: how many claims can be fully answerable with a set of fully faithful documents?

We label a claim as answerable if all of its sub-questions are answerable. Otherwise, a claim is partially answerable if at least one sub-question is partially answerable. We apply the same principle to compute claim-level faithfulness. Table 7.6 shows the results by combining the two factors. We see that addressing every aspect of complex claims is still challenging: 36 out of 50 claims contain at least one unanswerable question. For claims that can be fully addressed (all questions are either answerable or partially answerable), we see only 1 out of 14 contain a major

factual error in the retrieved documents.

7.6 Chapter Summary

In this chapter, we simulate realistic fact-checking scenarios by building a fact-checking pipeline that contains five components: claim decomposer, evidence retriever, evidence synthesizer, and veracity classifier. We show that the decomposed sub-questions are essential to retrieve good evidence to fact-check the claim and such evidence substantially improves the veracity classification performance. Also, through a human study, we show the GPT3-based evidence synthesizer generates faithful summaries of documents most of the time indicating it can be used as an effective part of the pipeline. Finally, we show that performance is bottlenecked by web retrieval and a human-machine-in-the-loop system might help retrieve better evidence.

Chapter 8

Future Directions

In this chapter, we discuss a few future directions motivated by existing works in this dissertation, including building reliable QA models in the context of foundation models and human-in-the-loop fact-checking.

8.1 Reliable QA systems with LLMs

In this section, we outline future research directions for improving the reliability of QA systems based on Large Language Models (LLMs), such as GPT-3 (Brown et al., 2020). Despite their impressive performance on a wide range of QA tasks, as discussed in Chapter 1, LLMs still make simple mistakes such as failing to match correct entities (Figure 1.2) and generate subtle hallucinations in longer answers. Enhancing the reliability of these models is both important and challenging.

Modifying the neural network structure, as demonstrated in Chapter 4, is impractical for LLMs with hundreds of billions of parameters. Instead, we draw inspiration from the answer verifier approach in Chapter 5 and propose using LLMs to self-critique their own answers. Recent work (Madaan et al., 2023; Paul et al., 2023; Huang et al., 2022) has shown that LLMs can refine their answers and achieve better performance using self-generated feedback or explanations. Nonetheless, several fundamental questions remain unanswered:

- Are LLMs more effective at critiquing their own answers than generating them in the first place?

- How do LLMs compare with humans in generating answer critiques?
- Can LLMs follow human-designed answer critiquing guidelines to generate explainable critiques?
- What critiquing guidelines are needed for different tasks and do they differ with different models?
- Can LLM-generated critiques be used to correct answers, create new "hard" examples, and further fine-tune LLMs for improved performance?

We aim to address these research questions in our future work, with a focus on developing more reliable and accurate QA systems that leverage the strengths of LLMs.

8.2 Human-in-the-loop Fact-checking

Despite the advancements explored in Chapter 6 and Chapter 7, evidence retrieval remains a challenging task for fact checking political claims due to two main obstacles: (1) there is no information available on the web and checking those claims involves direct communications with specific people or entities; (2) the sub-questions we generated are irrelevant or not properly formed to retrieve good evidence.

To address these challenges, we propose a human-in-the-loop fact-checking system as illustrated in figure 8.1. This system begins with the automated pipeline presented in Chapter 7, which provides fact-checkers with summarized documents and judgments. If the fact-checkers deem these documents unsatisfactory, the system reveals the sub-questions utilized for evidence retrieval, allowing fact-checkers to modify or create new questions. The system then employs the revised questions to retrieve additional documents, generating updated summaries and judgments. This

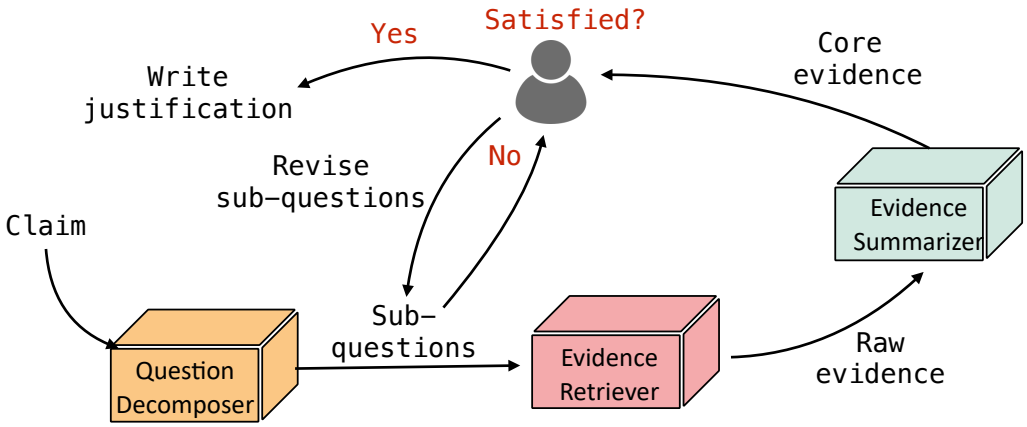


Figure 8.1: An overview of the human-in-the-loop fact-checking system where the human receives the outputs from the system and provides feedback.

iterative process continues until the fact-checkers are satisfied with the retrieved evidence.

Moreover, the system is able to further learn from the fact-check feedback to improve itself: for example, we can know what questions are important to retrieve good evidence and what questions are not according to the fact-checker and the system can learn from this signal.

Bibliography

Naser Ahmadi, Joohyung Lee, Paolo Papotti, and Mohammed Saeed. Explainable fact checking with probabilistic answer set programming. In *Conference on Truth and Trust Online*, 2019.

Chris Alberti, Daniel Andor, Emily Pitler, Jacob Devlin, and Michael Collins. Synthetic QA corpora generation with roundtrip consistency. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6168–6173, Florence, Italy, July 2019a. Association for Computational Linguistics. doi: 10.18653/v1/P19-1620. URL https://www.aclweb.org/anthology/P19-1620.

Chris Alberti, Kenton Lee, and Michael Collins. A BERT baseline for the Natural Questions. *arXiv preprint arXiv:1901.08634*, 2019b.

Tariq Alhindi, Savvas Petridis, and Smaranda Muresan. Where is your evidence: Improving fact-checking by justification modeling. In *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*, pages 85–90, Brussels, Belgium, November 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-5513. URL https://www.aclweb.org/anthology/W18-5513.

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

2021a. Association for Computational Linguistics. doi: 10.18653/v1/2021.fever-1.1. URL https://aclanthology.org/2021.fever-1.1.

Rami Aly, Zhijiang Guo, Michael Sejr Schlichtkrull, James Thorne, Andreas Vlachos, Christos Christodoulopoulos, Oana Cocarascu, and Arpit Mittal. FEVEROUS: Fact Extraction and VERification Over Unstructured and Structured information. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*, 2021b.

Moustafa Alzantot, Yash Sharma, Ahmed Elgohary, Bo-Jhang Ho, Mani Srivastava, and Kai-Wei Chang. Generating natural language adversarial examples. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2890–2896, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1316. URL https://www.aclweb.org/anthology/D18-1316.

Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. Neural module networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 39–48, 2016.

Akari Asai and Eunsol Choi. Challenges in information-seeking QA: Unanswerable questions and paragraph retrieval. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1492–1504, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.118. URL https://aclanthology.org/2021.acl-long.118.

Pepa Atanasova, Jakob Grue Simonsen, Christina Lioma, and Isabelle Augenstein. Generating fact checking explanations. In *Proceedings of the 58th Annual Meeting*

of the Association for Computational Linguistics, pages 7352–7364, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.656. URL https://aclanthology.org/2020.acl-main.656.

Isabelle Augenstein, Christina Lioma, Dongsheng Wang, Lucas Chaves Lima, Casper Hansen, Christian Hansen, and Jakob Grue Simonsen. 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, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1475. URL https://www.aclweb.org/anthology/D19-1475.

Sriram Balasubramanian, Naman Jain, Gaurav Jindal, Abhijeet Awasthi, and Sunita Sarawagi. What's in a name? are BERT named entity representations just as good for any other name? In *Proceedings of the 5th Workshop on Representation Learning for NLP*, pages 205–214, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.repl4nlp-1.24. URL https://www.aclweb.org/anthology/2020.repl4nlp-1.24.

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. Abstract meaning representation for sembanking. In *Proceedings of the 7th linguistic annotation workshop and interoperability with discourse*, pages 178–186, 2013.

Gagan Bansal, Tongshuang Wu, Joyce Zhou, Raymond Fok, Besmira Nushi, Ece Kamar, Marco Tulio Ribeiro, and Daniel Weld. Does the Whole Exceed Its Parts? The Effect of AI Explanations on Complementary Team Performance.

In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, CHI '21, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450380966. doi: 10.1145/3411764.3445717. URL https://doi.org/10.1145/3411764.3445717.

Max Bartolo, Alastair Roberts, Johannes Welbl, Sebastian Riedel, and Pontus Stenetorp. Beat the ai: Investigating adversarial human annotation for reading comprehension. *Transactions of the Association for Computational Linguistics*, 8:662–678, 2020.

Max Bartolo, Tristan Thrush, Robin Jia, Sebastian Riedel, Pontus Stenetorp, and Douwe Kiela. Improving question answering model robustness with synthetic adversarial data generation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8830–8848, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.696. URL https://aclanthology.org/2021.emnlp-main.696.

Yonatan Belinkov and Yonatan Bisk. Synthetic and natural noise both break neural machine translation. In *International Conference on Learning Representations*, 2018.

Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

Samuel Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. A large annotated corpus for learning natural language inference. In *Proceedings*

of the 2015 Conference on Empirical Methods in Natural Language Processing, pages 632–642, 2015.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language Models are Few-Shot Learners. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Yu Cao, Meng Fang, and Dacheng Tao. Bag: Bi-directional attention entity graph convolutional network for multi-hop reasoning question answering. *NAACL*, 2019.

Jifan Chen and Greg Durrett. Understanding dataset design choices for multi-hop reasoning. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 2019.

Jifan Chen and Greg Durrett. Robust question answering through sub-part alignment. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1251–1263, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.98. URL https://aclanthology.org/2021.naacl-main.98.

Jifan Chen, Shih-ting Lin, and Greg Durrett. Multi-hop question answering via reasoning chains. *arXiv preprint arXiv:1910.02610*, 2019a.

Jifan Chen, Eunsol Choi, and Greg Durrett. Can NLI models verify QA systems' predictions? In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3841–3854, Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-emnlp.324. URL https://aclanthology.org/2021.findings-emnlp.324.

Jifan Chen, Aniruddh Sriram, Eunsol Choi, and Greg Durrett. Generating literal and implied subquestions to fact-check complex claims. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3495–3516, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. URL https://aclanthology.org/2022.emnlp-main.229.

Jifan Chen, Grace Kim, Aniruddh Sriram, Greg Durrett, and Eunsol Choi. Complex claim verification with evidence retrieved in the wild. *arXiv preprint arXiv:2305.11859*, 2023.

Wenhu Chen, Hongmin Wang, Jianshu Chen, Yunkai Zhang, Hong Wang, Shiyang Li, Xiyou Zhou, and William Yang Wang. TabFact: A Large-scale Dataset for Table-based Fact Verification. In *International Conference on Learning Representations*, 2019b.

Eunsol Choi, Daniel Hewlett, Jakob Uszkoreit, Illia Polosukhin, Alexandre Lacoste, and Jonathan Berant. Coarse-to-fine question answering for long documents. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 209–220, 2017.

Eunsol Choi, Jennimaria Palomaki, Matthew Lamm, Tom Kwiatkowski, Dipanjan Das, and Michael Collins. Decontextualization: Making sentences stand-alone. *Transactions of the Association for Computational Linguistics*, 9:447–461, 2021. doi: 10.1162/tacl_a_00377. URL https://aclanthology.org/2021.tacl-1.27.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*, 2022.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*, 2022.

Volkan Cirik, Taylor Berg-Kirkpatrick, and Louis-Philippe Morency. Using syntax to ground referring expressions in natural images. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

Christopher Clark, Mark Yatskar, and Luke Zettlemoyer. Don't take the easy way out: Ensemble based methods for avoiding known dataset biases. 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 4060–4073, 2019.

Jonathan H. Clark, Eunsol Choi, Michael Collins, Dan Garrette, Tom Kwiatkowski, Vitaly Nikolaev, and Jennimaria Palomaki. TyDi QA: A benchmark for information-seeking question answering in typologically diverse languages. *Transactions of the Association for Computational Linguistics*, 8:454–470, 2020. doi: 10.1162/tacl_a_00317. URL https://www.aclweb.org/anthology/2020.tacl-1.30.

Jacob Cohen. A coefficient of agreement for nominal scales. *Educational and psychological measurement*, 20(1):37–46, 1960.

Silviu Cucerzan and Eugene Agichtein. Factoid question answering over unstructured and structured web content. In *Proceedings of the Fourteenth Text Retrieval Conference (TREC)*, 2005.

Ido Dagan, Oren Glickman, and Bernardo Magnini. The PASCAL Recognising Textual Entailment Challenge. In *Machine Learning Challenges Workshop*, pages 177–190. Springer, 2005.

Rajarshi Das, Shehzaad Dhuliawala, Manzil Zaheer, and Andrew McCallum. Multi-step retriever-reader interaction for scalable open-domain question answering. *arXiv preprint arXiv:1905.05733*, 2019.

Nicola De Cao, Wilker Aziz, and Ivan Titov. Question answering by reasoning across documents with graph convolutional networks. *EMNLP*, 2018.

Dorottya Demszky, Kelvin Guu, and Percy Liang. Transforming question answering datasets into natural language inference datasets. *arXiv preprint arXiv:1809.02922*, 2018.

Shrey Desai and Greg Durrett. Calibration of pre-trained transformers. *arXiv preprint arXiv:2003.07892*, 2020.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://www.aclweb.org/anthology/N19-1423.

Bhuwan Dhingra, Hanxiao Liu, Zhilin Yang, William W Cohen, and Ruslan Salakhutdinov. Gated-attention readers for text comprehension. *ACL*, 2016.

Xinya Du, Junru Shao, and Claire Cardie. 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, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1123. URL https://www.aclweb.org/anthology/P17-1123.

Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. DROP: A Reading Comprehension Benchmark Requiring Discrete Reasoning Over Paragraphs. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2368–2378, 2019.

Dheeru Dua, Cicero Nogueira dos Santos, Patrick Ng, Ben Athiwaratkun, Bing Xiang, Matt Gardner, and Sameer Singh. Generative context pair selection for multi-hop question answering. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7009–7015, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.561. URL https://aclanthology.org/2021.emnlp-main.561.

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

2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1090. URL https://www.aclweb.org/anthology/D17-1090.

Esin Durmus, He He, and Mona Diab. FEQA: A question answering evaluation framework for faithfulness assessment in abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5055–5070, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.454. URL https://www.aclweb.org/anthology/2020.acl-main.454.

Javid Ebrahimi, Daniel Lowd, and Dejing Dou. On adversarial examples for character-level neural machine translation. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 653–663, Santa Fe, New Mexico, USA, August 2018a. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/C18-1055.

Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. Hotflip: White-box adversarial examples for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 31–36, 2018b.

Tobias Falke, Leonardo FR Ribeiro, Prasetya Ajie Utama, Ido Dagan, and Iryna Gurevych. Ranking generated summaries by correctness: An interesting but challenging application for natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2214–2220, 2019.

Angela Fan, Aleksandra Piktus, Fabio Petroni, Guillaume Wenzek, Marzieh Saeidi, Andreas Vlachos, Antoine Bordes, and Sebastian Riedel. Generating fact checking briefs. In *Proceedings of the 2020 Conference on Empirical Methods in Natural*

Language Processing (EMNLP), pages 7147–7161, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.580. URL https://www.aclweb.org/anthology/2020.emnlp-main.580.

Yuwei Fang, Siqi Sun, Zhe Gan, Rohit Pillai, Shuohang Wang, and Jingjing Liu. Hierarchical graph network for multi-hop question answering. *arXiv preprint arXiv:1911.03631*, 2019.

Yair Feldman and Ran El-Yaniv. Multi-hop paragraph retrieval for open-domain question answering. *arXiv preprint arXiv:1906.06606*, 2019.

William Ferreira and Andreas Vlachos. Emergent: a novel data-set for stance classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1163–1168, San Diego, California, June 2016. Association for Computational Linguistics. doi: 10.18653/v1/N16-1138. URL https://aclanthology.org/N16-1138.

Adam Fisch, Alon Talmor, Robin Jia, Minjoon Seo, Eunsol Choi, and Danqi Chen. MRQA 2019 shared task: Evaluating generalization in reading comprehension. In *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*, pages 1–13, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-5801. URL https://www.aclweb.org/anthology/D19-5801.

Joseph L Fleiss. Measuring nominal scale agreement among many raters. *Psychological bulletin*, 76(5):378, 1971.

Saadia Gabriel, Skyler Hallinan, Maarten Sap, Pemi Nguyen, Franziska Roesner, Eunsol Choi, and Yejin Choi. Misinfo reaction frames: Reasoning about readers' reactions to news headlines. In *ACL*, 2022.

Mohamed H Gad-Elrab, Daria Stepanova, Jacopo Urbani, and Gerhard Weikum. Exfakt: A framework for explaining facts over knowledge graphs and text. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, pages 87–95, 2019.

Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F. Liu, Matthew Peters, Michael Schmitz, and Luke Zettlemoyer. AllenNLP: A deep semantic natural language processing platform. In *Proceedings of Workshop for NLP Open Source Software (NLP-OSS)*, pages 1–6, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-2501. URL https://www.aclweb.org/anthology/W18-2501.

Matt Gardner, Yoav Artzi, Victoria Basmov, Jonathan Berant, Ben Bogin, Sihao Chen, Pradeep Dasigi, Dheeru Dua, Yanai Elazar, Ananth Gottumukkala, Nitish Gupta, Hannaneh Hajishirzi, Gabriel Ilharco, Daniel Khashabi, Kevin Lin, Jiangming Liu, Nelson F. Liu, Phoebe Mulcaire, Qiang Ning, Sameer Singh, Noah A. Smith, Sanjay Subramanian, Reut Tsarfaty, Eric Wallace, Ally Zhang, and Ben Zhou. Evaluating models' local decision boundaries via contrast sets. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1307–1323, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.findings-emnlp.117. URL https://www.aclweb.org/anthology/2020.findings-emnlp.117.

Matt Gardner, William Merrill, Jesse Dodge, Matthew E. Peters, Alexis Ross, Sameer Singh, and Noah A. Smith. Competency Problems: On Finding and Removing Artifacts in Language Data. *ArXiv*, abs/2104.08646, 2021.

Ameya Godbole, Dilip Kavarthapu, Rajarshi Das, Zhiyu Gong, Abhishek Singhal, Hamed Zamani, Mo Yu, Tian Gao, Xiaoxiao Guo, Manzil Zaheer, et al. Multi-

step entity-centric information retrieval for multi-hop question answering. *arXiv preprint arXiv:1909.07598*, 2019.

Lucas Graves. Understanding the Promise and Limits of Automated Fact-Checking. Technical report, Reuters Institute, University of Oxford, 2018.

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

Ashim Gupta and Vivek Srikumar. X-fact: A new benchmark dataset for multilingual fact checking. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 675–682, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-short.86. URL https://aclanthology.org/2021.acl-short.86.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. REALM: Retrieval-augmented language model pre-training. *arXiv preprint arXiv:2002.08909*, 2020.

Andreas Hanselowski, Christian Stab, Claudia Schulz, Zile Li, and Iryna Gurevych. 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, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/K19-1046. URL https://www.aclweb.org/anthology/K19-1046.

Sanda Harabagiu and Andrew Hickl. Methods for using textual entailment in open-domain question answering. In *Proceedings of the 21st International Conference*

on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics, pages 905–912, Sydney, Australia, July 2006. Association for Computational Linguistics. doi: 10.3115/1220175.1220289. URL https://www.aclweb.org/anthology/P06-1114.

Luheng He, Mike Lewis, and Luke Zettlemoyer. Question-answer driven semantic role labeling: Using natural language to annotate natural language. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 643–653, Lisbon, Portugal, September 2015. Association for Computational Linguistics. doi: 10.18653/v1/D15-1076. URL https://www.aclweb.org/anthology/D15-1076.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. DeBERTa: Decoding-enhanced BERT with Disentangled Attention. In *International Conference on Learning Representations*, 2020.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations*, 2020.

Karl Moritz Hermann, Tomás Kociskỳ, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. Teaching machines to read and comprehend. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2015.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The Curious Case of Neural Text Degeneration. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2020. URL https://arxiv.org/abs/1904.09751.

Minghao Hu, Yuxing Peng, Zhen Huang, Xipeng Qiu, Furu Wei, and Ming Zhou. Reinforced mnemonic reader for machine reading comprehension. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pages 4099–4106. AAAI Press, 2018.

Minghao Hu, Furu Wei, Yuxing Peng, Zhen Huang, Nan Yang, and Dongsheng Li. Read+ verify: Machine reading comprehension with unanswerable questions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6529–6537, 2019.

Ronghang Hu, Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Kate Saenko. Learning to reason: End-to-end module networks for visual question answering. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 804–813, 2017.

Jiaxin Huang, Shixiang Shane Gu, Le Hou, Yuexin Wu, Xuezhi Wang, Hongkun Yu, and Jiawei Han. Large language models can self-improve. *arXiv preprint arXiv:2210.11610*, 2022.

Drew A Hudson and Christopher D Manning. Compositional attention networks for machine reasoning. *arXiv preprint arXiv:1803.03067*, 2018.

Sathish Reddy Indurthi, Seunghak Yu, Seohyun Back, and Heriberto Cuayáhuitl. Cut to the chase: A context zoom-in network for reading comprehension. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 570–575, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1054. URL https://www.aclweb.org/anthology/D18-1054.

Mohit Iyyer, John Wieting, Kevin Gimpel, and Luke Zettlemoyer. Adversarial example generation with syntactically controlled paraphrase networks. 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 1875–1885, 2018.

Sarthak Jain and Byron C. Wallace. Attention is not explanation. In *NAACL*, 2019.

Sarthak Jain, Sarah Wiegreffe, Yuval Pinter, and Byron C Wallace. Learning to faithfully rationalize by construction. *arXiv preprint arXiv:2005.00115*, 2020.

Peter A Jansen, Elizabeth Wainwright, Steven Marmorstein, and Clayton T Morrison. Worldtree: A corpus of explanation graphs for elementary science questions supporting multi-hop inference. *LREC*, 2018.

Robin Jia. *Building robust natural language processing systems*. Stanford University, 2020.

Robin Jia and Percy Liang. Adversarial examples for evaluating reading comprehension systems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2021–2031, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1215. URL https://www.aclweb.org/anthology/D17-1215.

Robin Jia, Aditi Raghunathan, Kerem Göksel, and Percy Liang. Certified robustness to adversarial word substitutions. 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 4129–4142, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1423. URL https://www.aclweb.org/anthology/D19-1423.

Yichen Jiang and Mohit Bansal. Self-assembling modular networks for interpretable multi-hop reasoning. 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 4464–4474, 2019.

Yichen Jiang, Nitish Joshi, Yen-Chun Chen, and Mohit Bansal. Explore, propose, and assemble: An interpretable model for multi-hop reading comprehension. *arXiv preprint arXiv:1906.05210*, 2019.

Yichen Jiang, Shikha Bordia, Zheng Zhong, Charles Dognin, Maneesh Singh, and Mohit Bansal. HoVer: A dataset for many-hop fact extraction and claim verification. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3441–3460, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.findings-emnlp.309. URL https://www.aclweb.org/anthology/2020.findings-emnlp.309.

Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. Is bert really robust? a strong baseline for natural language attack on text classification and entailment. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 8018–8025, 2020.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, Vancouver, Canada, July 2017a. Association for Computational Linguistics. doi: 10.18653/v1/P17-1147. URL https://www.aclweb.org/anthology/P17-1147.

Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. TriviaQA: A Large Scale Distantly Supervised Challenge Dataset for Reading Comprehension.

In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, Vancouver, Canada, July 2017b. Association for Computational Linguistics.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S Weld, Luke Zettlemoyer, and Omer Levy. Spanbert: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77, 2020.

Amita Kamath, Robin Jia, and Percy Liang. Selective question answering under domain shift. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5684–5696, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.503. URL https://www.aclweb.org/anthology/2020.acl-main.503.

Ryo Kamoi, Tanya Goyal, and Greg Durrett. Shortcomings of Question Answering Based Factuality Frameworks for Error Localization. In *arXiv*, 2022.

Divyansh Kaushik, Eduard Hovy, and Zachary Lipton. Learning the difference that makes a difference with counterfactually-augmented data. In *International Conference on Learning Representations*, 2019.

Aniruddha Kembhavi, Minjoon Seo, Dustin Schwenk, Jonghyun Choi, Ali Farhadi, and Hannaneh Hajishirzi. Are you smarter than a sixth grader? textbook question answering for multimodal machine comprehension. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4999–5007, 2017.

Daniel Khashabi, Snigdha Chaturvedi, Michael Roth, Shyam Upadhyay, and Dan Roth. Looking beyond the surface: A challenge set for reading comprehension over

multiple sentences. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, volume 1, pages 252–262, 2018a.

Daniel Khashabi, Tushar Khot, Ashish Sabharwal, and Dan Roth. Question answering as global reasoning over semantic abstractions. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018b.

Daniel Khashabi, Tushar Khot, and Ashish Sabharwal. More bang for your buck: Natural perturbation for robust question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 163–170, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.12. URL https://www.aclweb.org/anthology/2020.emnlp-main.12.

Tushar Khot, Ashish Sabharwal, and Peter Clark. SciTail: A textual entailment dataset from science question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

Douwe Kiela, Max Bartolo, Yixin Nie, Divyansh Kaushik, Atticus Geiger, Zhengxuan Wu, Bertie Vidgen, Grusha Prasad, Amanpreet Singh, Pratik Ringshia, Zhiyi Ma, Tristan Thrush, Sebastian Riedel, Zeerak Waseem, Pontus Stenetorp, Robin Jia, Mohit Bansal, Christopher Potts, and Adina Williams. Dynabench: Rethinking benchmarking in NLP. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4110–4124, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.324. URL https://aclanthology.org/2021.naacl-main.324.

Najoung Kim, Ellie Pavlick, Burcu Karagol Ayan, and Deepak Ramachandran. Which linguist invented the lightbulb? presupposition verification for question-answering. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3932–3945, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.304. URL https://aclanthology.org/2021.acl-long.304.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *ICLR*, 2017.

Ayal Klein, Jonathan Mamou, Valentina Pyatkin, Daniela Stepanov, Hangfeng He, Dan Roth, Luke Zettlemoyer, and Ido Dagan. QANom: Question-answer driven SRL for nominalizations. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3069–3083, Barcelona, Spain (Online), December 2020. International Committee on Computational Linguistics. URL https://www.aclweb.org/anthology/2020.coling-main.274.

Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, et al. Wilds: A benchmark of in-the-wild distribution shifts. In *International Conference on Machine Learning*, pages 5637–5664. PMLR, 2021.

Neema Kotonya and Francesca Toni. Explainable automated fact-checking for public health claims. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7740–7754, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/

v1/2020.emnlp-main.623. URL https://www.aclweb.org/anthology/2020.emnlp-main.623.

Ankit Kumar, Ozan Irsoy, Peter Ondruska, Mohit Iyyer, James Bradbury, Ishaan Gulrajani, Victor Zhong, Romain Paulus, and Richard Socher. Ask me anything: Dynamic memory networks for natural language processing. In *International Conference on Machine Learning*, pages 1378–1387, 2016.

Souvik Kundu, Tushar Khot, and Ashish Sabharwal. Exploiting explicit paths for multi-hop reading comprehension. *arXiv preprint arXiv:1811.01127*, 2018.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466, March 2019. doi: 10.1162/tacl_a_00276. URL https://www.aclweb.org/anthology/Q19-1026.

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. RACE: Large-scale ReAding Comprehension Dataset From Examinations. *arXiv preprint arXiv:1704.04683*, 2017.

Matthew Lamm, Jennimaria Palomaki, Chris Alberti, Daniel Andor, Eunsol Choi, Livio Baldini Soares, and Michael Collins. Qed: A framework and dataset for explanations in question answering. *arXiv preprint arXiv:2009.06354*, 2020.

Dong Bok Lee, Seanie Lee, Woo Tae Jeong, Donghwan Kim, and Sung Ju Hwang. Generating diverse and consistent QA pairs from contexts with information-maximizing hierarchical conditional VAEs. In *Proceedings of the 58th Annual*

Meeting of the Association for Computational Linguistics, pages 208–224, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.20. URL https://www.aclweb.org/anthology/2020.acl-main.20.

Kenton Lee, Luheng He, Mike Lewis, and Luke Zettlemoyer. End-to-end neural coreference resolution. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 188–197, 2017.

Mike Lewis and Angela Fan. Generative question answering: Learning to answer the whole question. In *International Conference on Learning Representations*, 2018.

Haoran Li, Junnan Zhu, Jiajun Zhang, and Chengqing Zong. Ensure the correctness of the summary: Incorporate entailment knowledge into abstractive sentence summarization. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1430–1441, 2018.

Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, et al. Holistic evaluation of language models. *arXiv preprint arXiv:2211.09110*, 2022.

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

Kai Liu, Xin Liu, An Yang, Jing Liu, Jinsong Su, Sujian Li, and Qiaoqiao She. A robust adversarial training approach to machine reading comprehension. In *AAAI*, pages 8392–8400, 2020.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. RoBERTa: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

Yi-Ju Lu and Cheng-Te Li. GCAN: Graph-aware co-attention networks for explainable fake news detection on social media. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 505–514, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.48. URL https://www.aclweb.org/anthology/2020.acl-main.48.

Zhiyi Ma, Kawin Ethayarajh, Tristan Thrush, Somya Jain, Ledell Wu, Robin Jia, Christopher Potts, Adina Williams, and Douwe Kiela. Dynaboard: An evaluation-as-a-service platform for holistic next-generation benchmarking. *Advances in Neural Information Processing Systems*, 34:10351–10367, 2021.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. Self-refine: Iterative refinement with self-feedback. *arXiv preprint arXiv:2303.17651*, 2023.

Bodhisattwa Prasad Majumder, Sudha Rao, Michel Galley, and Julian McAuley. Ask what's missing and what's useful: Improving clarification question generation using global knowledge. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4300–4312, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.340. URL https://aclanthology.org/2021.naacl-main.340.

Sewon Min, Victor Zhong, Richard Socher, and Caiming Xiong. Efficient and robust question answering from minimal context over documents. *ACL*, 2018.

Sewon Min, Eric Wallace, Sameer Singh, Matt Gardner, Hannaneh Hajishirzi, and Luke Zettlemoyer. Compositional questions do not necessitate multi-hop reasoning. In *ACL*, 2019a.

Sewon Min, Victor Zhong, Luke Zettlemoyer, and Hannaneh Hajishirzi. Multi-hop reading comprehension through question decomposition and rescoring. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6097–6109, Florence, Italy, July 2019b. Association for Computational Linguistics. doi: 10.18653/v1/P19-1613. URL https://www.aclweb.org/anthology/P19-1613.

Anshuman Mishra, Dhruvesh Patel, Aparna Vijayakumar, Xiang Lorraine Li, Pavan Kapanipathi, and Kartik Talamadupula. Looking beyond sentence-level natural language inference for question answering and text summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1322–1336, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.104. URL https://aclanthology.org/2021.naacl-main.104.

Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, Xu Jiang, Karl Cobbe, Tyna Eloundou, Gretchen Krueger, Kevin Button, Matthew Knight, Benjamin Chess, and John Schulman. WebGPT: Browser-assisted question-answering with human feedback. *arXiv*, 2112.09332, 2022.

Preslav Nakov, David Corney, Maram Hasanain, Firoj Alam, Tamer Elsayed, Alberto Barr'on-Cedeno, Paolo Papotti, Shaden Shaar, and Giovanni Da San Martino. Automated fact-checking for assisting human fact-checkers. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence (IJCAI)*, 2021.

Yixin Nie, Haonan Chen, and Mohit Bansal. Combining fact extraction and ver-

ification with neural semantic matching networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6859–6866, 2019.

Kosuke Nishida, Kyosuke Nishida, Masaaki Nagata, Atsushi Otsuka, Itsumi Saito, Hisako Asano, and Junji Tomita. Answering while summarizing: Multi-task learning for multi-hop QA with evidence extraction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2335–2345, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1225. URL https://www.aclweb.org/anthology/P19-1225.

Wojciech Ostrowski, Arnav Arora, Pepa Atanasova, and Isabelle Augenstein. Multi-hop fact checking of political claims. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence (IJCAI)*, 2021.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *arXiv preprint arXiv:2203.02155*, 2022.

Martha Palmer, Daniel Gildea, and Paul Kingsbury. The Proposition Bank: An Annotated Corpus of Semantic Roles. *Comput. Linguist.*, 31(1):71–106, March 2005. ISSN 0891-2017. doi: 10.1162/0891201053630264. URL https://doi.org/10.1162/0891201053630264.

Ramakanth Pasunuru, Han Guo, and Mohit Bansal. Towards improving abstractive summarization via entailment generation. In *Proceedings of the Workshop on New Frontiers in Summarization*, pages 27–32, 2017.

Debjit Paul, Mete Ismayilzada, Maxime Peyrard, Beatriz Borges, Antoine Bosselut, Robert West, and Boi Faltings. Refiner: Reasoning feedback on intermediate representations. *arXiv preprint arXiv:2304.01904*, 2023.

Ethan Perez, Patrick Lewis, Wen-tau Yih, Kyunghyun Cho, and Douwe Kiela. Unsupervised question decomposition for question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8864–8880, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.713. URL https://www.aclweb.org/anthology/2020.emnlp-main.713.

Verónica Pérez-Rosas, Bennett Kleinberg, Alexandra Lefevre, and Rada Mihalcea. Automatic detection of fake news. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3391–3401, Santa Fe, New Mexico, USA, August 2018. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/C18-1287.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. 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 2227–2237, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1202. URL https://www.aclweb.org/anthology/N18-1202.

Aleksandra Piktus, Necati Bora Edizel, Piotr Bojanowski, Edouard Grave, Rui Ferreira, and Fabrizio Silvestri. Misspelling oblivious word embeddings. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3226–3234, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1326. URL https://www.aclweb.org/anthology/N19-1326.

Kashyap Popat, Subhabrata Mukherjee, Jannik Strötgen, and Gerhard Weikum. Where the truth lies: Explaining the credibility of emerging claims on the web and social media. In *Proceedings of the 26th International Conference on World Wide Web Companion*, pages 1003–1012. International World Wide Web Conferences Steering Committee, 2017.

Kashyap Popat, Subhabrata Mukherjee, Andrew Yates, and Gerhard Weikum. 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, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1003. URL https://www.aclweb.org/anthology/D18-1003.

Valentina Pyatkin, Ayal Klein, Reut Tsarfaty, and Ido Dagan. QADiscourse - Discourse Relations as QA Pairs: Representation, Crowdsourcing and Baselines. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2804–2819, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.224. URL https://www.aclweb.org/anthology/2020.emnlp-main.224.

Peng Qi, Xiaowen Lin, Leo Mehr, Zijian Wang, and Christopher D Manning. Answering complex open-domain questions through iterative query generation. *arXiv preprint arXiv:1910.07000*, 2019.

Lin Qiu, Yunxuan Xiao, Yanru Qu, Hao Zhou, Lei Li, Weinan Zhang, and Yong Yu. Dynamically fused graph network for multi-hop reasoning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6140–6150, Florence, Italy, July 2019. Association for Computational Linguis-

tics. doi: 10.18653/v1/P19-1617. URL https://www.aclweb.org/anthology/P19-1617.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020a. URL http://jmlr.org/papers/v21/20-074.html.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21:1–67, 2020b.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 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, Austin, Texas, November 2016a. Association for Computational Linguistics. doi: 10.18653/v1/D16-1264. URL https://www.aclweb.org/anthology/D16-1264.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 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, 2016b.

Pranav Rajpurkar, Robin Jia, and Percy Liang. Know what you don't know: Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789, Melbourne, Australia, July 2018. Association for Computational Linguis-

tics. doi: 10.18653/v1/P18-2124. URL https://www.aclweb.org/anthology/P18-2124.

Sudha Rao and Hal Daumé III. 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, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1255. URL https://www.aclweb.org/anthology/P18-1255.

Hannah Rashkin, Eunsol Choi, Jin Yea Jang, Svitlana Volkova, and Yejin Choi. Truth of varying shades: Analyzing language in fake news and political fact-checking. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2931–2937, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1317. URL https://aclanthology.org/D17-1317.

Siva Reddy, Danqi Chen, and Christopher D. Manning. CoQA: A conversational question answering challenge. *Transactions of the Association for Computational Linguistics*, 7:249–266, March 2019. doi: 10.1162/tacl_a_00266. URL https://www.aclweb.org/anthology/Q19-1016.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Semantically equivalent adversarial rules for debugging NLP models. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 856–865, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1079. URL https://www.aclweb.org/anthology/P18-1079.

Matthew Richardson, Christopher J.C. Burges, and Erin Renshaw. MCTest: A challenge dataset for the open-domain machine comprehension of text. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 193–203, Seattle, Washington, USA, October 2013. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/D13-1020.

Arkadiy Saakyan, Tuhin Chakrabarty, and Smaranda Muresan. COVID-fact: Fact extraction and verification of real-world claims on COVID-19 pandemic. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2116–2129, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.165. URL https://aclanthology.org/2021.acl-long.165.

Mrinmaya Sachan and Eric Xing. Machine comprehension using rich semantic representations. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 486–492, 2016.

Mrinmaya Sachan and Eric Xing. Self-training for jointly learning to ask and answer questions. 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 629–640, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1058. URL https://www.aclweb.org/anthology/N18-1058.

Mrinmaya Sachan, Kumar Dubey, Eric Xing, and Matthew Richardson. Learning answer-entailing structures for machine comprehension. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the*

7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 239–249, 2015.

Shiori Sagawa, Pang Wei Koh, Tony Lee, Irena Gao, Sang Michael Xie, Kendrick Shen, Ananya Kumar, Weihua Hu, Michihiro Yasunaga, H Marklund, et al. Extending the wilds benchmark for unsupervised adaptation. In *International Conference on Learning Representations*, 2022.

Tal Schuster, Adam Fisch, and Regina Barzilay. Get Your Vitamin C! Robust Fact Verification with Contrastive Evidence. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 624–643, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.52. URL https://aclanthology.org/2021.naacl-main.52.

Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. Bidirectional attention flow for machine comprehension. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2017.

Yelong Shen, Po-Sen Huang, Jianfeng Gao, and Weizhu Chen. Reasonet: Learning to stop reading in machine comprehension. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1047–1055. ACM, 2017.

Peng Shi and Jimmy Lin. Simple bert models for relation extraction and semantic role labeling. *arXiv preprint arXiv:1904.05255*, 2019.

Kai Shu, Limeng Cui, Suhang Wang, Dongwon Lee, and Huan Liu. defend: Explainable fake news detection. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 395–405, 2019.

Vered Shwartz, Peter West, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Unsupervised commonsense question answering with self-talk. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4615–4629, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.373. URL https://www.aclweb.org/anthology/2020.emnlp-main.373.

Prakhar Singh, Anubrata Das, Junyi Jessy Li, and Matthew Lease. The Case for Claim Difficulty Assessment in Automatic Fact Checking. *arXiv ePrint 2109.09689*, 2021a. URL https://arxiv.org/abs/2109.09689.

Prakhar Singh, Anubrata Das, Junyi Jessy Li, and Matthew Lease. The case for claim difficulty assessment in automatic fact checking. *arXiv preprint arXiv:2109.09689*, 2021b.

Linfeng Song, Zhiguo Wang, Mo Yu, Yue Zhang, Radu Florian, and Daniel Gildea. Exploring Graph-structured Passage Representation for Multi-hop Reading Comprehension with Graph Neural Networks. *arXiv preprint arXiv:1809.02040*, 2018.

Asher Stern and Ido Dagan. A confidence model for syntactically-motivated entailment proofs. In *Proceedings of the International Conference Recent Advances in Natural Language Processing 2011*, pages 455–462, Hissar, Bulgaria, September 2011. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/R11-1063.

Sainbayar Sukhbaatar, Jason Weston, Rob Fergus, et al. End-to-end memory networks. In *Advances in neural information processing systems*, pages 2440–2448, 2015.

Swabha Swayamdipta, Roy Schwartz, Nicholas Lourie, Yizhong Wang, Hannaneh Hajishirzi, Noah A. Smith, and Yejin Choi. Dataset Cartography: Mapping and

Diagnosing Datasets with Training Dynamics. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020.

Alon Talmor and Jonathan Berant. The web as a knowledge-base for answering complex questions. 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 641–651, 2018.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 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, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1074. URL https://www.aclweb.org/anthology/N18-1074.

Tristan Thrush, Kushal Tirumala, Anmol Gupta, Max Bartolo, Pedro Rodriguez, Tariq Kane, William Gaviria Rojas, Peter Mattson, Adina Williams, and Douwe Kiela. Dynatask: A framework for creating dynamic AI benchmark tasks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 174–181, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-demo.17. URL https://aclanthology.org/2022.acl-demo.17.

Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordoni, Philip Bachman, and Kaheer Suleman. Newsqa: A machine comprehension dataset. *ACL 2017*, page 191, 2017.

Harsh Trivedi, Heeyoung Kwon, Tushar Khot, Ashish Sabharwal, and Niranjan Balasubramanian. Repurposing Entailment for Multi-Hop Question Answering

Tasks. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 2019.

George Tsatsaronis, Georgios Balikas, Prodromos Malakasiotis, Ioannis Partalas, Matthias Zschunke, Michael R Alvers, Dirk Weissenborn, Anastasia Krithara, Sergios Petridis, Dimitris Polychronopoulos, Yannis Almirantis, John Pavlopoulos, Nicolas Baskiotis, Patrick Gallinari, Thierry Artieres, Axel Ngonga, Norman Heino, Eric Gaussier, Liliana Barrio-Alvers, Michael Schroeder, Ion Androutsopoulos, and Georgios Paliouras. An overview of the bioasq large-scale biomedical semantic indexing and question answering competition. *BMC Bioinformatics*, 16, 2015.

Sebastian Tschiatschek, Adish Singla, Manuel Gomez-Rodriguez, Arpit Merchant, and Andreas Krause. Fake news detection in social networks via crowd signals. In *The Web Conference, Alternate Track on Journalism, Misinformation, and Fact-checking*, 2018.

Ming Tu, Kevin Huang, Guangtao Wang, Jing Huang, Xiaodong He, and Bowen Zhou. Select, answer and explain: Interpretable multi-hop reading comprehension over multiple documents. *arXiv preprint arXiv:1911.00484*, 2019a.

Ming Tu, Guangtao Wang, Jing Huang, Yun Tang, Xiaodong He, and Bowen Zhou. Multi-hop reading comprehension across multiple documents by reasoning over heterogeneous graphs. *arXiv preprint arXiv:1905.07374*, 2019b.

Prasetya Ajie Utama, Nafise Sadat Moosavi, and Iryna Gurevych. Towards debiasing nlu models from unknown biases. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7597–7610, 2020.

Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. Pointer networks. In *Advances in Neural Information Processing Systems*, pages 2692–2700, 2015.

Andreas Vlachos and Sebastian Riedel. Fact Checking: Task definition and dataset construction. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2014a.

Andreas Vlachos and Sebastian Riedel. 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, June 2014b. Association for Computational Linguistics. doi: 10.3115/v1/W14-2508. URL https://www.aclweb.org/anthology/W14-2508.

Svitlana Volkova, Kyle Shaffer, Jin Yea Jang, and Nathan Hodas. Separating facts from fiction: Linguistic models to classify suspicious and trusted news posts on Twitter. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 647–653, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-2102. URL https://aclanthology.org/P17-2102.

David Wadden, Shanchuan Lin, Kyle Lo, Lucy Lu Wang, Madeleine van Zuylen, Arman Cohan, and Hannaneh Hajishirzi. Fact or fiction: Verifying scientific claims. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7534–7550, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.609. URL https://www.aclweb.org/anthology/2020.emnlp-main.609.

Eric Wallace, Shi Feng, Nikhil Kandpal, Matt Gardner, and Sameer Singh. Universal adversarial triggers for attacking and analyzing NLP. 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 2153–2162, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1221. URL https://www.aclweb.org/anthology/D19-1221.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium, November 2018a. Association for Computational Linguistics. doi: 10.18653/v1/W18-5446. URL https://www.aclweb.org/anthology/W18-5446.

Alex Wang, Kyunghyun Cho, and Mike Lewis. Asking and answering questions to evaluate the factual consistency of summaries. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5008–5020, Online, July 2020a. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.450. URL https://www.aclweb.org/anthology/2020.acl-main.450.

Chao Wang and Hui Jiang. Explicit utilization of general knowledge in machine reading comprehension. *arXiv preprint arXiv:1809.03449*, 2018.

Hai Wang, Dian Yu, Kai Sun, Jianshu Chen, Dong Yu, Dan Roth, and David McAllester. Evidence sentence extraction for machine reading comprehension. *arXiv preprint arXiv:1902.08852*, 2019.

Shuohang Wang, Mo Yu, Xiaoxiao Guo, Zhiguo Wang, Tim Klinger, Wei Zhang, Shiyu Chang, Gerry Tesauro, Bowen Zhou, and Jing Jiang. R 3: Reinforced

ranker-reader for open-domain question answering. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018b.

William Yang Wang. "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, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-2067. URL https://www.aclweb.org/anthology/P17-2067.

Xuguang Wang, Linjun Shou, Ming Gong, Nan Duan, and Daxin Jiang. No answer is better than wrong answer: A reflection model for document level machine reading comprehension. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4141–4150, Online, November 2020b. Association for Computational Linguistics. doi: 10.18653/v1/2020.findings-emnlp.370. URL https://www.aclweb.org/anthology/2020.findings-emnlp.370.

Yicheng Wang and Mohit Bansal. Robust machine comprehension models via adversarial training. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 575–581, 2018.

Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. Finetuned language models are zero-shot learners. *ICLR*, 2021.

Johannes Welbl, Pontus Stenetorp, and Sebastian Riedel. Constructing datasets for multi-hop reading comprehension across documents. *Transactions of the Association of Computational Linguistics*, 6:287–302, 2018.

Sean Welleck, Jason Weston, Arthur Szlam, and Kyunghyun Cho. Dialogue natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3731–3741, 2019.

Jason Weston, Antoine Bordes, Sumit Chopra, Alexander M Rush, Bart van Merriënboer, Armand Joulin, and Tomas Mikolov. Towards AI-Complete Question Answering: A set of prerequisite toy tasks. *arXiv preprint arXiv:1502.05698*, 2015.

Adina Williams, Nikita Nangia, and Samuel Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana, June 2018a. Association for Computational Linguistics. doi: 10.18653/v1/N18-1101. URL https://www.aclweb.org/anthology/N18-1101.

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

Tomer Wolfson, Mor Geva, Ankit Gupta, Matt Gardner, Yoav Goldberg, Daniel Deutch, and Jonathan Berant. Break it down: A question understanding benchmark. *Transactions of the Association for Computational Linguistics*, 8:183–198, 2020.

Jun Yan, Yang Xiao, Sagnik Mukherjee, Bill Yuchen Lin, Robin Jia, and Xiang Ren. On the robustness of reading comprehension models to entity renaming. In

Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 508–520, Seattle, United States, July 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.naacl-main.37. URL https://aclanthology.org/2022.naacl-main.37.

Fan Yang, Shiva K Pentyala, Sina Mohseni, Mengnan Du, Hao Yuan, Rhema Linder, Eric D Ragan, Shuiwang Ji, and Xia Hu. Xfake: Explainable fake news detector with visualizations. In *The World Wide Web Conference*, pages 3600–3604, 2019a.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W Cohen, Ruslan Salakhutdinov, and Christopher D Manning. HotpotQA: A Dataset for Diverse, Explainable Multi-hop Question Answering. *EMNLP*, 2018.

Ziqing Yang, Yiming Cui, Wanxiang Che, Ting Liu, Shijin Wang, and Guoping Hu. Improving machine reading comprehension via adversarial training. *arXiv preprint arXiv:1911.03614*, 2019b.

Yi-Ting Yeh and Yun-Nung Chen. Qainfomax: Learning robust question answering system by mutual information maximization. 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 3361–3366, 2019.

Wenpeng Yin, Nazneen Fatema Rajani, Dragomir Radev, Richard Socher, and Caiming Xiong. Universal natural language processing with limited annotations: Try few-shot textual entailment as a start. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8229–8239, Online, November 2020. Association for Computational Linguistics. doi:

10.18653/v1/2020.emnlp-main.660. URL https://www.aclweb.org/anthology/2020.emnlp-main.660.

Wenpeng Yin, Dragomir Radev, and Caiming Xiong. DocNLI: A large-scale dataset for document-level natural language inference. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4913–4922, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-acl.435. URL https://aclanthology.org/2021.findings-acl.435.

Michael Zhang and Eunsol Choi. SituatedQA: Incorporating extra-linguistic contexts into QA. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7371–7387, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.586. URL https://aclanthology.org/2021.emnlp-main.586.

Shujian Zhang, Chengyue Gong, and Eunsol Choi. Knowing more about questions can help: Improving calibration in question answering. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1958–1970, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-acl.172. URL https://aclanthology.org/2021.findings-acl.172.

Victor Zhong, Caiming Xiong, Nitish Shirish Keskar, and Richard Socher. Coarse-grain fine-grain coattention network for multi-evidence question answering. *ICLR*, 2019.

Mantong Zhou, Minlie Huang, and Xiaoyan Zhu. Robust reading comprehension with linguistic constraints via posterior regularization. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28:2500–2510, 2020.

Yimeng Zhuang and Huadong Wang. Token-level dynamic self-attention network for multi-passage reading comprehension. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2252–2262, 2019.

Vita

Jifan Chen was born and brought up in Mengzi, Yunnan, China. He earned his bachelor's and master's degrees in science from the Department of Computer Science at Fudan University. In 2023, he successfully completed his Ph.D. in computer science from the University of Texas at Austin. His research primarily focuses on constructing modular and interpretable NLP systems, such as question answering and fact-checking. His notable contributions have been published in prestigious conferences and journals, including ACL, EMNLP, NAACL, and AAAI. Additionally, during his doctoral studies, Jifan Chen gained valuable industry experience through summer internships at Google and Amazon AWS.