Can NLI Models Verify QA Systems' Predictions?

Jifan Chen **Eunsol Choi** **Greg Durrett**
Department of Computer Science
The University of Texas at Austin
{jfchen, eunsol, gdurrett}@cs.utexas.edu

Abstract

To build robust question answering systems, we need the ability to verify whether answers to questions are truly correct, not just "good enough" in the context of imperfect QA datasets. We explore the use of natural language inference (NLI) as a way to achieve this goal, as NLI inherently requires the premise (document context) to contain all necessary information to support the hypothesis (proposed answer to the question). We leverage large pretrained 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, i.e., when the answer sentence does not address all aspects of the question.

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 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).

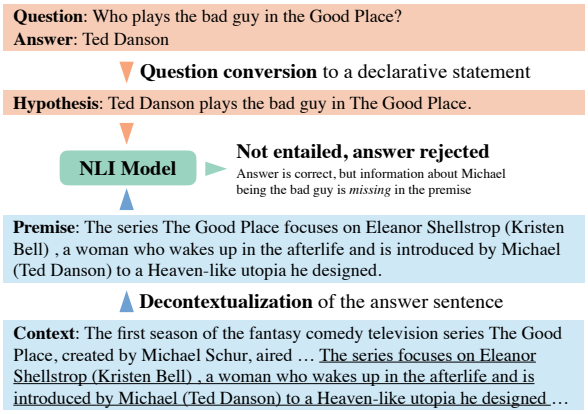


Figure 1: An example from the Natural Questions dataset demonstrating how to convert a (question, context, answer) triplet to a (premise, hypothesis) pair. The underlined text denotes the sentence containing the answer *Ted Danson*, which is then decontextualized by replacing *The series* with *The series 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., 2018; 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., 2018) 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

Findings of the Association for Computational Linguistics: EMNLP 2021, pages 3841–3854
November 7–11, 2021.

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 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., 2020). 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 improve calibration of QA models or boost their confidence in their decisions? Second, how does the entailment criterion of NLI, which is defined somewhat coarsely by crowd annotators (Williams et al., 2018), 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 calibration over the base QA model across five different datasets.**[1] 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 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.

[1]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

2 Using NLI as a QA Verifier

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., 2018).

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?

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., 2016), for which we can get an answer candidate by running a pre-trained QA model.[2] We use the `Bert-joint` model proposed by Alberti et al. (2019) 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 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., 2020), 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., 2018; 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.

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

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 Appendix B.

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., 2017), 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.[3]

[3]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 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).

Base QA Model We train our base QA model (Alberti et al., 2019) 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., 2018, 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 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.

Some basic statistics for each dataset after processing with our pipeline are shown in Appendix A.

4 Improving QA Calibration 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.

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 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).

4.2 Calibration

To analyze the effectiveness of the NLI models in a more systematic way, we test whether they can improve calibration of QA models or 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.

4.2.1 Comparison Systems

NLI model variants We train separate NLI models with MNLI, NQ-NLI, NQ-NLI+MNLI intro-

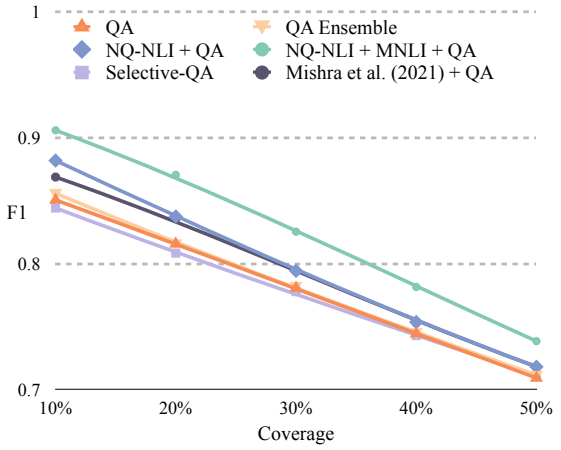


Figure 3: Average calibration performance of our models *combining the posterior* from the NQ-NLI and the QA models over five datasets. The x-axis denotes the top $k\%$ of examples the model is answering, ranked by the confidence score. The y-axis denotes the F1 score.

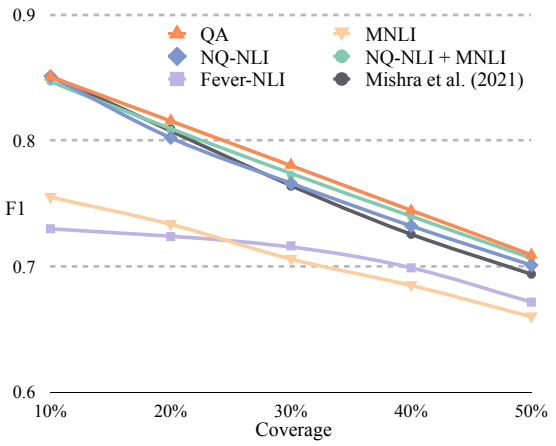


Figure 4: Average calibration performance of our NLI models *alone* (not including QA posteriors) trained on NQ-NLI over five datasets. The x-axis denotes the top $k\%$ of examples the model is answering, ranked by the confidence score. The y-axis denotes the F1 score.

duced in Section 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 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 model.

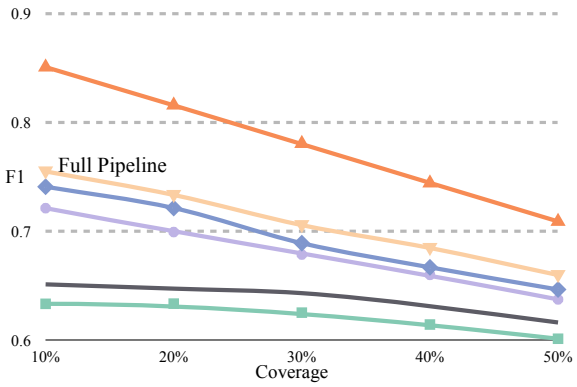
4.2.2 Results and Analysis

Figure 3 shows the macro-averaged results over the five QA datasets. Please refer to Appendix C 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 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 in Appendix C, 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'



	Premise	Hypothesis
QA	N/A	N/A
MNLI	Decontextualized	Converted
MNLI-DO	Decontextualized	Original
MNLI-SC	Sentence	Converted
MNLI-SO	Sentence	Original
MNLI-FC	Full-context	Converted

Figure 5: Average calibration 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.

posteriors is necessary for this strong performance. Figure 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 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 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 shows the average results across five datasets and Figure 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 6.2.

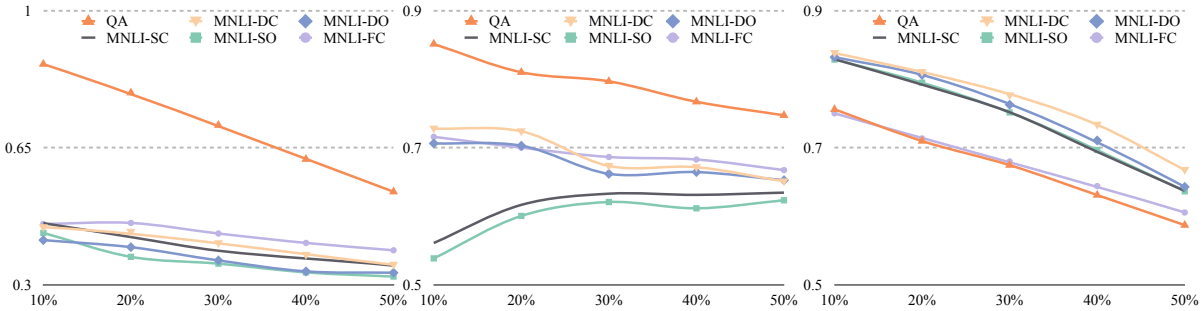
Moreover, we see that MNLI outperforms the base QA posteriors on SQuAD2.0 and SQuAD-adv. Figure 6(a) also shows that the largest gap between the QA and NLI model is on NQ, which is unsurprising since the QA model is trained on NQ. These results show how the improvement in the last section is achieved: the complementary strengths of MNLI and NQ datasets lead to the best overall performance.

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 6.1 and 6.2. All of the authors of this paper 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 1.

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 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



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

Figure 6: Calibration 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. The x-axis denotes coverage and the y-axis denotes the F1 score.

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

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 7, the model fails to replace *The work* with its full entity name *The Art of War*.

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 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 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

	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 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.

are missing or the annotated answer is wrong. From Table 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 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.

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, 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 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., 2020; 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 ex-

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 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.

plicit 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.

8 Conclusion

This work presents a strong pipeline for converting QA examples into NLI examples, with the intent of verifying 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.

Acknowledgments

This work was partially supported by NSF Grant IIS-1814522. We would like to thank Kaj Bostrom, Yasumasa Onoe, and the anonymous reviewers for their helpful comments.

This material is also based on research that is in part supported by the Air Force Research Laboratory (AFRL), DARPA, for the KAIROS program under agreement number FA8750-19-2-1003. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Air Force Research Laboratory (AFRL), DARPA, or the U.S. Government.

References

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

Akari Asai and Eunsol Choi. 2021. 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 (Vol-*

ume 1: Long Papers), pages 1492–1504, Online. Association for Computational Linguistics.

Samuel Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. 2015. 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.

Jifan Chen and Greg Durrett. 2019. 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)*.

Jifan Chen and Greg Durrett. 2021. Robust question answering through sub-part alignment. *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Eunsol Choi, Jennimaria Palomaki, Matthew Lamm, Tom Kwiatkowski, Dipanjan Das, and Michael Collins. 2021. Decontextualization: Making sentences stand-alone. *Transactions of the Association for Computational Linguistics*.

Christopher Clark, Mark Yatskar, and Luke Zettlemoyer. 2019. 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.

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

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

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

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. 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.

Tobias Falke, Leonardo FR Ribeiro, Prasetya Ajie Utama, Ido Dagan, and Iryna Gurevych. 2019. 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.

Adam Fisch, Alon Talmor, Robin Jia, Minjoon Seo, Eunsol Choi, and Danqi Chen. 2019. 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. Association for Computational Linguistics.

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. 2020. Evaluating models' local decision boundaries via contrast sets. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1307–1323, Online. Association for Computational Linguistics.

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

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

Sanda Harabagiu and Andrew Hickl. 2006. 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. Association for Computational Linguistics.

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

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

Robin Jia and Percy Liang. 2017. 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. Association for Computational Linguistics.

Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. 2017. TriviaQA: A Large Scale Distantly Supervised Challenge Dataset for Reading Comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, Vancouver, Canada. Association for Computational Linguistics.

Amita Kamath, Robin Jia, and Percy Liang. 2020. Selective question answering under domain shift. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5684–5696, Online. Association for Computational Linguistics.

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

Daniel Khashabi, Tushar Khot, and Ashish Sabharwal. 2020. 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.

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

Najoung Kim, Ellie Pavlick, Burcu Karagol Ayan, and Deepak Ramachandran. 2021. 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. Association for Computational Linguistics.

Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *ICLR (Poster)*.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. 2019. Natural Questions: A Benchmark for Question Answering Research. *Transactions of the Association for Computational Linguistics*, 7:453–466.

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

Haoran Li, Junnan Zhu, Jiajun Zhang, and Chengqing Zong. 2018. 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.

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

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

Anshuman Mishra, Dhruvesh Patel, Aparna Vijayakumar, Xiang Lorraine Li, Pavan Kapanipathi, and Kartik Talamadupula. 2021. 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. Association for Computational Linguistics.

Yixin Nie, Haonan Chen, and Mohit Bansal. 2019. Combining fact extraction and verification with neural semantic matching networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6859–6866.

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

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

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. 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.

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

Jeff Rasley, Samyam Rajbhandari, Olatunji Ruwase, and Yuxiong He. 2020. Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters. In *Proceedings of the 26th ACM SIGKDD International Conference on*

Knowledge Discovery & Data Mining, pages 3505–3506.

Matthew Richardson, Christopher J.C. Burges, and Erin Renshaw. 2013. 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. Association for Computational Linguistics.

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

Asher Stern and Ido Dagan. 2011. 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. Association for Computational Linguistics.

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

Harsh Trivedi, Heeyoung Kwon, Tushar Khot, Ashish Sabharwal, and Niranjan Balasubramanian. 2019. 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)*, pages 2948–2958.

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. 2015. An overview of the BIOASQ large-scale biomedical semantic indexing and question answering competition. *BMC Bioinformatics*, 16.

Eric Wallace, Shi Feng, Nikhil Kandpal, Matt Gardner, and Sameer Singh. 2019. 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. Association for Computational Linguistics.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. 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. Association for Computational Linguistics.

Xuguang Wang, Linjun Shou, Ming Gong, Nan Duan, and Daxin Jiang. 2020. 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. Association for Computational Linguistics.

Yicheng Wang and Mohit Bansal. 2018. 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.

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

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. 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.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Yi-Ting Yeh and Yun-Nung Chen. 2019. 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.

Wenpeng Yin, Nazneen Fatema Rajani, Dragomir Radev, Richard Socher, and Caiming Xiong. 2020. 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. Association for Computational Linguistics.

Shujian Zhang, Chengyue Gong, and Eunsol Choi. 2021. 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. Association for Computational Linguistics.

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

A Statistics of the Converted Datasets

The statistics of the datasets after processing through our pipeline is shown in Table 2. Both the premise length and the hypothesis length are quite similar except for the premise length of TriviaQA, despite their original context length differs greatly (Fisch et al., 2019).

	Prem Len	Hyp Len	Word Overlap
NQ	20.0	8.0	0.22
TriviaQA	15.9	9.0	0.16
BioASQ	20.6	8.0	0.14
SQuAD 2.0	19.1	8.2	0.23
SQuAD-adv	19.0	8.2	0.26

Table 2: Statistics of the development set for each dataset listed above. Here, "Prem len" and "Hyp len" denote the average number of words with stop words removed in the premise and hypothesis respectively; "Word Overlap" denotes the Jaccard similarity between the premise and the hypothesis.

B Model Details

B.1 Answer Generator

We train our `Bert-joint` on the full NQ training set for 1 epoch. We initialize the model with `bert-large-uncased-wwm`.[4] The batch size is set to 8, window size is set to 512, and the optimizer we use is Adam (Kingma and Ba, 2015) with initial learning rate setting to 3e-5.

B.2 Question Converter

Each instance of the input is constructed as [CLS]q[S]a[S], where [CLS] and [S] are the classification and separator tokens of the T5 model respectively. The output is the target sentence d.

The model is trained using the seq2seq framework of Huggingface (Wolf et al., 2020). The max source sequence length is set to 256 and the target sequence length is set to 512. Batch size is set to 12 and we use Deepspeed for memory optimization (Rasley et al., 2020). We train the model with 86k question-answer pairs for 1 epoch with Adam optimizer and an initial learning rate set to 3e-5. 95% of question answer pairs come from SQuAD and the remaining 5% come from four other question answering datasets (Demszky et al., 2018).

B.3 Decontextualizer

Each instance of the input is constructed as follows:

[CLS]T[S]$x_1, ..., x_{t-1}$[S]x_t[S]$x_{t+1}, ..., x_n$[S]

where [CLS] and [S] are the classification and separator tokens of the T5 model respectively. T denotes the context title which could be empty. x_i denotes the ith sentence in the context and x_t is the target sentence to decontextualize.

The model is trained using the seq2seq framework of Huggingface (Wolf et al., 2020). The max sequence length for both source and target is set to 512. Batch size is set to 4 and we use Deepspeed for memory optimization (Rasley et al., 2020). We train the model with 11k question-answer pairs (Choi et al., 2021) for 5 epoch with Adam optimizer and an initial learning rate set to 3e-5.

B.4 NQ-NLI

The generated NQ-NLI training and development set contain 191k and 4,855 (premise, hypothesis) pairs from NQ respectively. We initialize the model with `roberta-large` (Liu et al., 2019) and train the model for 5 epochs. Batch size is set to 16, with Adam as the optimizer and initial learning rate set to 2e-6.

C Performance Breakdown on All Datasets

Figures 9 and 10 show full results for Figures 4 and 3, respectively.

[4] https://github.com/google-research/bert

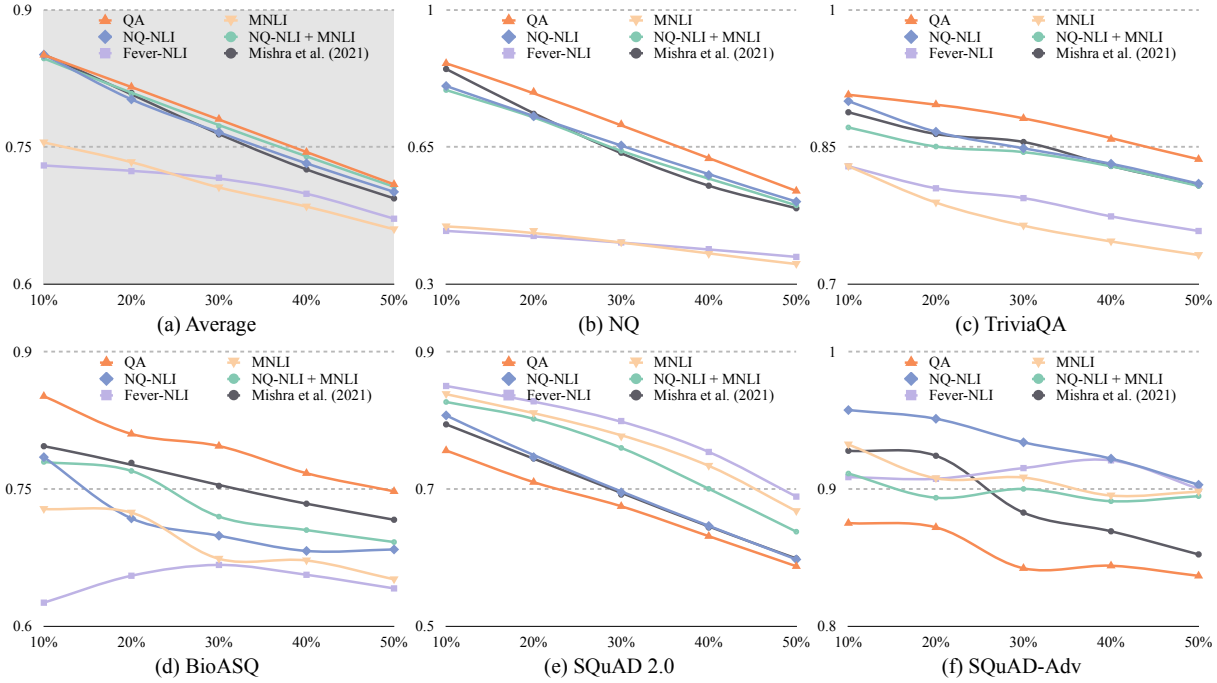


Figure 9: Calibration performance of the NQ-NLI models on five QA datasets we used in the paper. The training using NQ-NLI helps close the gap between the QA and the NLI models. The x-axis denotes coverage and the y-axis denotes the F1 score.

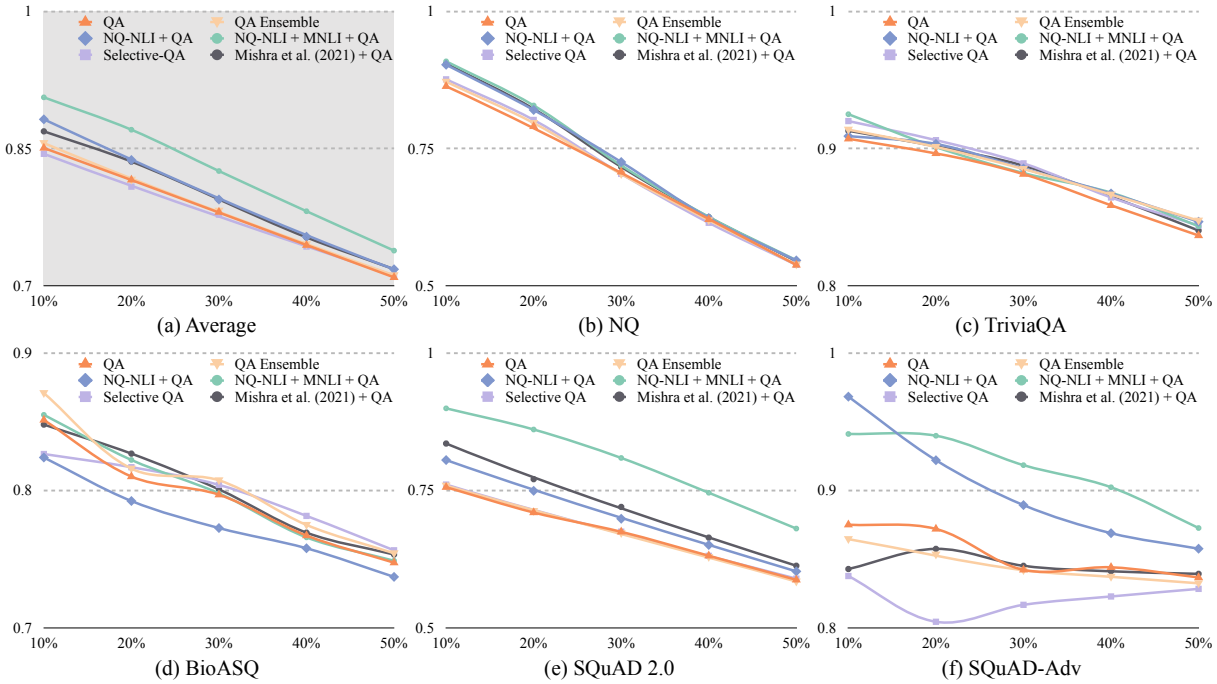


Figure 10: Calibration performance of the NQ-NLI models combined with the QA model on five QA datasets we used in the paper. The combined **NQ-NLI+MNLI+QA** model largely outperforms the QA model on all datasets. The x-axis denotes coverage and the y-axis denotes the F1 score.