Do We Know What We Don't Know?
Studying Unanswerable Questions beyond SQuAD 2.0

Elior Sulem, Jamaal Hay and Dan Roth
Department of Computer and Information Science, University of Pennsylvania
eliors,jamaalh,danroth@seas.upenn.edu

Abstract

Understanding when a text snippet does not provide a sought after information is an essential part of natural language understanding. Recent work (SQuAD 2.0, Rajpurkar et al., 2018) has attempted to make some progress in this direction by enriching the SQuAD dataset for the Extractive QA task with unanswerable questions. However, as we show, the performance of a top system trained on SQuAD 2.0 drops considerably in out-of-domain scenarios, limiting its use in practical situations. In order to study this we build an out-of-domain corpus, focusing on simple event-based questions and distinguish between two types of IDK questions: competitive questions, where the context includes an entity of the same type as the expected answer, and simpler, non-competitive questions where there is no entity of the same type in the context. We find that SQuAD 2.0-based models fail even in the case of the simpler questions. We then analyze the similarities and differences between the IDK phenomenon in Extractive QA and the Recognizing Textual Entailments task (RTE, Dagan et al., 2013) and investigate the extent to which the latter can be used to improve the performance.[1]

1 Introduction

Extractive Question Answering (Extractive QA) has attracted a lot of interest in recent years with the creation of large-scale datasets (Rajpurkar et al., 2016, 2018) and has seen large improvements with the use of contextualized language models such as BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019). However, the ability to extract information from text only addresses one aspect of the expectations we have from a comprehension system. Another main aspect concerns the ability to identify that a given information is not in the text, a witness of understanding in human comprehension.

Context: John was born in New York.

Q1: Where did John marry?

Answer: IDK - **Competitive**

Q2: When was John born?

Answer: IDK - **Non-competitive**

Figure 1: Examples of a competitive (Q1) and a non-competitive (Q2) IDK questions.

The ability to answer "IDK" allows one to address more realistic situations in reading comprehension, both as an end task and as an intermediary step for other NLP applications, such as QA-based event extraction (Chen et al., 2020; Lyu et al., 2021) or QA-based summarization evaluation (Deutsch et al., 2021).

To begin addressing this important phenomenon, Rajpurkar et al. (2018) added unanswerable questions to SQuAD 1.1 (Rajpurkar et al., 2016), providing a useful resource for identifying IDK cases in the Extactive QA case (SQuAD 2.0). However, as we show, the performance of a top system trained on SQuAD 2.0 considerably drops on out-of-domain simple questions.

In this paper, we show that SQuAD 2.0 alone is not sufficient to address IDK questions in practical situations. For this purpose, we introduce a new evaluation dataset of very simple questions on single-sentence contexts that we compile based on an event extraction corpus (ACE, Walker et al., 2006). In particular, we propose to separately eval-

[1] The new datasets along with all the other artifacts generated here are available at http://cogcomp.org/page/publication_view/955.

Findings of the Association for Computational Linguistics: EMNLP 2021, pages 4543–4548
November 7–11, 2021. ©2021 Association for Computational Linguistics

Corpus		Split	# Examples	IDK (%)	Task	Data Annotation
Existing Corpora						
MNLI		train	392,702	33	RTE	"entailment", "contradiction", "neutral"
		dev	9,815	32		
SQuAD 2.0		train	130,319	33	Extractive QA	extracted span, "[]"
		dev	11,873	50		
New Corpora						
ACE-whQA	Has answer	test	238	0	Extractive QA	extracted span, "[]"
	Compet. IDK	test	250	100		
	non-Compet. IDK	test	246	100		

Table 1: Statistics and properties of existing corpora we use (top) and the newly introduced corpus (bottom).

uate the performance of a QA model on two types of IDK questions: (i) cases where the context includes an entity of the same type as the expected answer such as Q1 in Figure 1 where “New York" is a location appearing in the context 1. We call this type of questions *competitive IDK questions* and (ii) cases where the context includes no entity of the same type as the expected answer such as Q2 in Figure 1 where the question expects a time mention while the context does not include time.

Evaluating on the new dataset, we find that a top SQuAD 2.0 model obtains low scores. even in the case of the simpler, non-competitive IDK questions, only reaching 28.46 F1 (Section 4).

We then explore the use of another Natural Language Understanding (NLU) task that also includes an IDK option. We focus on the Recognizing Textual Entailments (RTE, Dagan et al., 2013) task and find that leveraging it considerably improves the results in the case of non-competitive IDK questions but is not sufficient for reaching a good performance in the competitive IDK cases.

2 Related Work

Unanswerable questions have been first addressed in the context of the annual TREC competition for open-domain QA (Voorhees, 2002), where the subtask of span extraction has some similarities with Extractive QA, although in the former the goal is to answer a question from a large collection of documents. In Extractive QA, a system being able to answer "I don’t know" has been proposed by Levy et al. (2017) in the framework of the relation extraction task which is formulated in QA terms. Another example is the use of QA systems for event extraction, as recently proposed by Chen et al. (2020) who modified a BERT-based QA system to predict an argument role in a clozed test format. In this work we evaluate our Extract QA systems on event-based questions questions derived from the ACE corpus (Walker et al., 2006), focusing on the location and time argument types. We differ from Chen et al. (2020) by experimenting in an out-of-domain setting and by preserving the QA format. We also distinguish between easier IDK cases when there is no entities of the argument type expected by the question and harder cases where an entity of the same type appears in the sentence (see Section 3).

Rajpurkar et al. (2018) enriched the SQuAD 1.1 corpus by including unanswerable questions for the same paragraphs via crowdsourcing, resulting in SQuAD 2.0, that we are using in this paper for the Extractive QA task. We show that training on SQuAD 2.0 is not sufficient to address IDK in out-of-domain settings (focusing on simple, event-based questions) and that the RTE data can be useful to address a particular type of IDK questions.

Rajpurkar et al. (2018) experimented on SQuAD 2.0 using the BiDAF-No-Answer (BNA) model proposed by Levy et al. (2017) and the DocumentQA No-Answer (DocQA) model from Clark and Gardner (2018). These models learn to predict the probability that a question is unanswerable, in addition to a distribution over answer choices. This also holds in the BERT implementation we use here.

An alternative way for training and prediction in the case of unanswerable questions has been advanced by Tan et al. (2018) who proposed to first predict whether there is an answer in the context. Tan et al. (2018) also used a predict+validate approach, which is also explored by Hu et al. (2019) who added a separately trained answer verifier for no-answer detection. We do not modify the training and prediction used in the BERT paper approach but rather explore the performance in out-of-domain scenarios as well as the use of RTE to improve the performance.

The selective question answering task in out-of-domain settings (Kamath et al., 2020) is related to the identification of unanswerable questions. However, it targets the ability of a system to refrain from answering in some of the cases in order to avoid errors in out-of-domain settings, independently from the presence of the answer in the context. The au-

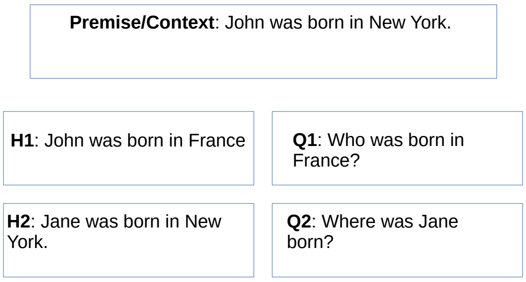

Premise/Context: John was born in New York.
H1: John was born in France
Q1: Who was born in France?
H2: Jane was born in New York.
Q2: Where was Jane born?

Figure 2: Examples of RTE hypotheses (left) and wh-questions (right) given a premise/context.

thors show that selective prediction methods do not identify unanswerable questions, suggesting that an explicit labeling of IDK in the training data is necessary in our case.

In the RTE task (Dagan et al., 2013), the IDK option is instantiated by the "neutral" category. In some of the RTE works (Bentivogli et al., 2009; Wang et al., 2018), “contradiction" and “neutral" are unified in a “non-entailed" joint category. Demszky et al. (2018) proposed a conversion of Extractive QA datasets to 2-label RTE format. We instead leverage the RTE task for Extractive QA via additional pretraining and compare between the presence and the absence of an IDK label in the RTE data (See Section 5).

3 Test Datasets

We leverage the ACE event extraction (Walker et al., 2006)[2] dataset to derive questions asking about the argument that participates in an event, given the trigger. This allow us to experiment on IDK answers that result from the fact that one of the event arguments is missing. For this purpose, we first select sentence fragments that include a location or a time mention according to the ACE annotation. To generate the wh-questions, we automatically generate candidate questions based on the event structure by asking both where and when did T happen, where T is the event trigger. The answer is labeled “I don’t know” when the entity type is missing.

To generate additional IDK questions, we select more sentences from the ACE dataset that do not necessarily include time/location mentions. All the questions are manually validated to ensure both grammatical and logical correctness. We compile two types of IDK questions. The first concerns IDK questions where there is an entity in the context that has the same type as the expected answer; this creates competition and makes the prediction harder (Compet. IDK). For creating this type of examples, we manually modify the context sentence to add an entity that has the same type as the expected answer. We choose the entity randomly from a set of time/locations entities appearing in the dataset. For example, given the context “She went to Mexico after she lost her seat in the 1997 election", a Compet. IDK question is “Where is the loss?". The second type of questions (non-Compet. IDK) concerns cases where there is no candidate of the same type in the sentence. In this case too, we use manual modifications. For example, given the context “He was arrested for his crimes", a non-compet. IDK question is “When was the arrest?". The resulting corpus, called ACE-whQA includes three sub-corpora: “Has Answer", “Compet. IDK" and “non-Compet. IDK" with 238, 250 and 246 examples respectively. More examples are presented in Figure 3.

4 Training on SQuAD 2.0 is Not Sufficient

We finetune the BERT-LARGE-CASED representation on the SQuAD 2.0 dataset and evaluate on ACE-whQA.[3] We also report the score on the SQuAD 2.0 dev set (80.96 F1).

The evaluation on the ACE-whQA dataset is presented in The first column of Table 3. We find that for "Has Answer" the performance of the baseline trained on SQuAD 2.0 drops, compared to the in-domain setting but still achieves acceptable performance. However, in the case of IDK, we observe that even in the case of easy questions, with no competition from an entity of the same type (non-Compet. IDK), the performance of the baseline system is very low (28.46).

5 Exploring the use of the RTE task

Similarities and Differences The Recognizing Textual Entailment (RTE) task (Dagan et al., 2013) consists of classifying a sentence pair composed of a premise p and a hypothesis h into three classes, according to the relation between the two sentences: “entailment", “contradiction" and “neutral", which corresponds to the IDK option. Although the instances of IDK in RTE and Extractive QA share

[2] https://catalog.ldc.upenn.edu/LDC2006T06

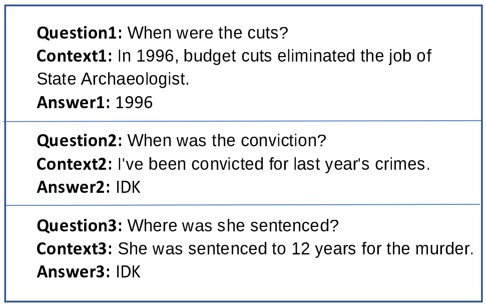
[3] For training on SQuAD 2.0, we use two train epochs and fine-tune for the learning rate (3e-5 and 5e-5) and the batch size (24 and 48).

Train → Test ↓	SQuAD 2.0	MNLI + SQuAD 2.0	c(MNLI) + SQuAD 2.0
All	80.96	81.92*	**82.60***
Has answer	83.53	**84.63**	84.12
IDK	78.40	79.23*	**81.09***

Table 2: F1 scores of the different systems, **tested on SQuAD 2.0 Dev for the Extractive QA task.** The rows represent the training strategies. The columns represent the test datasets. In all the cases the trained representation is BERT-LARGE-CASED. In each line the highest score is presented in bold. The scores significantly higher (using a one-sided t-test, $p < 0.05$) than the baseline (the first column) appear with a star (*).

Train → Test ↓	SQuAD 2.0	MNLI + SQuAD 2.0	c(MNLI)+ SQuAD 2.0
Has answer	62.39	71.68	**78.13**
Compet. IDK	20.8	**46.40***	26.00
non-Compet. IDK	28.46	**75.61***	47.15$^{*\circ}$

Table 3: F1 scores of the different systems, **tested on the ACE-whQA out-of-domain test set for the Extractive QA task**. In all the cases the trained representation is BERT-LARGE-CASED. In each line the highest score is presented in bold. The scores significantly higher (using a one-sided t-test, $p < 0.05$) than the baseline (the first column) appear with a star (*). Scores that are significantly higher than the baseline and in the same time, significantly lower than the top system, are presented with a circle ($^\circ$).



Question1: When were the cuts?
Context1: In 1996, budget cuts eliminated the job of State Archaeologist.
Answer1: 1996

Question2: When was the conviction?
Context2: I've been convicted for last year's crimes.
Answer2: IDK

Question3: Where was she sentenced?
Context3: She was sentenced to 12 years for the murder.
Answer3: IDK

Figure 3: Examples of (1) Has-answer, (2) Competitive IDK and (3) Non-competitive questions from the ACE-whQA dataset.

a common idea, there are also considerable differences. First, the format of a wh-question is missing some content which is already present in a corresponding RTE hypothesis; for example, the location entity in a “where" question. Therefore, a wh-question cannot be directly converted to an RTE hypothesis, independently from the context. This format difference is also related to the fact that RTE can be seen as a classification task, while Extract QA involves span extraction. Second, a conversion between the formats will not always preserve the IDK label, as illustrated in H1 and Q1 in Figure 2. In particular, an IDK instance in Extractive QA can correspond also to a "contradiction" in RTE. Finally, while short paragraphs are used in SQuAD 2.0, the premises in the MNLI corpus for the RTE task are single sentences. While this is not inherent in the definition of the respective tasks, the available datasets impact the models used by the community.

Experimental Setting Here we consider Extractive QA as a target task. RTE is the auxiliary task. Our baseline system consists in the BERT-LARGE-CASED representation fine-tuned on the SQuAD 2.0 train corpus. We experiment with the following systems: (i) **MNLI + SQuAD 2.0** where we first finetune BERT-LARGE on MNLI, remove the classification layer and further finetune on SQuAD 2.0. (ii) c(MNLI) **+ SQuAD 2.0**: 2-label pretraining on MNLI, where we only consider the "contradiction" and "non-contradiction" classes.[4] In all cases we evaluate the system on the SQuAD 2.0 dev as well as the three sub-corpora of ACE-whQA introduced in Section 3: questions that have an answer (Has answer), questions that do not have an answer but there is an entity in the sentence of the same type as the expected answer (Competitive IDK) and questions that do not have an answer and there is no entity of the same type (non-competitive. IDK).

For training on MNLI with the BERT-LARGE-CASED representation, we use batch size of 32 and 3 training epochs. We fine tune over three possible learning rate values: 2e-5, 3e-5 and 5e-5. For training on SQuAD 2.0, we use the same hyperparameters as in Section 4. For each of the training settings, we choose the hyperparameter combination that maximizes the accuracy for the

[4] We chose this binary version for the experiments (the other versions being "entailment"/"non-entailment" and "neutral"/"non-neutral") since it achieved the highest score on the corresponding binary MNLI dev set (92.50 accuracy).

target task on the SQuAD 2.0 dev set.

Results The evaluation on the SQuAD 2.0 dev set is presented in Table 2, where we report the F1 scores. We observe that the use of MNLI for additional pretraining is helpful, siginificantly improving both the overall and the IDK scores[5] △ SQuAD 2.0 where the additional pretraining is done on the binary MNLI train corpus, which achieves the best performance but is not significantly better than the use of the 3-label MNLI.

The evaluation on the ACE-whQA dataset is presented in Table 3. We find that for "Has Answer" the performance of the baseline trained on SQuAD 2.0 drops, compared to the in-domain setting but still achieves acceptable performance. The best performance is obtained where c(MNLI) is used for pretraining, reaching an F1 score of 78.13. However, in the case of IDK, we observe that even in the case of easy questions, with no competition from an entity of the same type (non-Compet. IDK), the performance of the baseline system is very low (28.46). The use of MNLI for additional pretraining greatly improves the performance, achieving an F1 score of 75.61. For the harder IDK questions (where there is an entity of the same type in the context), the performance significantly improves as well when using MNLI ($p < 0.05$) but it only reaches a score of 46.40, leaving room for additional research.

We also observe that the best model in the in-domain setting that uses the binary MNLI corpus (with the same amount of data), achieves low results on IDK cases (and significantly lower with respect to the 3-label MNLI) showing the importance of training on the three labels to address event-based IDK questions.

6 Conclusion

We studied the IDK phenomenon, which is essential in language comprehension, in Extractive QA, going beyond the evaluation on SQuAD 2.0. We designed an out-of-domain evaluation dataset, composed of two main types of IDK questions. We show that IDK in Extractive QA is a major challenge for current NLP systems. We further explore the use of the RTE dataset and observe a considerable improvement in the case of non-competitive questions. Future work concerns the use of additional Natural Language Understanding tasks and data for IDK and the improvement of the ability to face adversarial IDK questions.

Acknowledgements

We thank Qing Lyu and the members of the Cognitive Computation Group for their insightful feedback ad well as the reviewers of the paper for their useful suggestions. This work was supported by Contracts FA8750-19-2-1004 and FA8750-19-2-0201 with the US Defense Advanced Research Projects Agency (DARPA). Approved for Public Release, Distribution Unlimited. This research is also based upon work supported in part by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via IARPA Contract No. 2019-19051600006 under the BETTER Program. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of ODNI, IARPA, the Department of Defense, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation therein. This material is also based upon work supported by Google Cloud (TFRC Program).

References

L. Bentivogli, I. Dagan, H. T. Dang, D. Giampiccolo, and B. Magnini. 2009. The fifth PASCAL recognizing textual entailment challenge. In *Proc. of TAC Workshop*.

Yunmo Chen, Tongfei Chen, Seth Ebner, and Benjamin Van Durme. 2020. Reading the manual: Event extraction as definition comprehension. In *Proceedings of the Fourth Workshop on Structured Prediction for NLP*, pages 74–83.

Christopher Clark and Matt Gardner. 2018. Simple and effective multi-paragraph reading comprehension. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

Ido Dagan, Dan Roth, Mark Sammons, and Fabio Massimo Zanzoto. 2013. Recognizing Textual Entailment: Models and Applications.

Dorottya Demszky, Kelvin Guu, and Percy Liang. 2018. Transforming question answering datasets into natural language inference datasets. ArXiv:1809.02922 [cs.CL].

[5] one-sided t-test, $p < 0.05$

Daniel Deutsch, Tania Bedrax-Weiss, and Dan Roth. 2021. Towards question-answering as an automatic metric for evaluating the content quality of a summary. *TACL*, 9:774–789.

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.

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*, page 6529–6537.

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.

Omer Levy, Minjoon Seo, Eunsol Choi, and Luke Zettlemoyer. 2017. Zero-shot relation extraction via reading comprehension. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 333–342.

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.

Qing Lyu, Hongming Zhang, Elior Sulem, and Dan Roth. 2021. Zero-shot event extraction via transfer learning: Challenges and insights. In *ACL 2021*, page 322–332.

Pranav Rajpurkar, Robin Jia, and D. 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.

Chuanqi Tan, Furu Wei, Qingyu Zhou, Nan Yang, Weifeng Lv, and Ming Zhou. 2018. I know there is no answer: Modeling answer validation for machine reading comprehension. In *Natural Language Processing and Chinese Computing*, pages 85–97.

E. Voorhees. 2002. Overview of the TREC-2002 question answering track. In *The Eleventh TREC Conference*.

Christopher Walker, Stephanie Strassel, Julie Medero, and Kazuaki Maeda. 2006. Ace 2005 multilingual training corpus. Linguistic Data Consortium, Philadelphia, 57.

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.