A Pipeline for Generating, Annotating and Employing Synthetic Data for Real World Question Answering

Matt Maufe
Filament AI, UK
University of Warwick, UK
matt.maufe@filament.ai

James Ravenscroft
Filament AI, UK
University of Warwick, UK
james.ravenscroft@filament.ai

Rob Procter
University of Warwick, UK
The Alan Turing Institute, UK
Rob.Procter@warwick.ac.uk

Maria Liakata
Queen Mary University of London, UK
The Alan Turing Institute, UK
m.liakata@qmul.ac.uk

Abstract

Question Answering (QA) is a growing area of research, often used to facilitate the extraction of information from within documents. State-of-the-art QA models are usually pre-trained on domain-general corpora like Wikipedia and thus tend to struggle on out-of-domain documents without fine-tuning. We demonstrate that synthetic domain-specific datasets can be generated easily using domain-general models, while still providing significant improvements to QA performance. We present two new tools for this task: A flexible pipeline for validating the synthetic QA data and training downstream models on it, and an online interface to facilitate human annotation of this generated data. Using this interface, crowdworkers labelled 1117 synthetic QA pairs, which we then used to fine-tune downstream models and improve domain-specific QA performance by 8.75 F1.

1 Introduction

Having enough relevant training data is a key factor for achieving strong performance in machine learning and NLP (Hoffmann et al., 2022), but for many tasks, large domain-specific datasets are expensive and time-consuming to create manually. This is especially true for tasks like Extractive Question Answering (QA), which both relies on domain-specific knowledge and requires skilled annotators. These difficulties have led to increased interest in synthetic data generation recently (Feng et al., 2021) through various methods such as bootstrapping from smaller datasets, or through generative models which create entirely new data.

We make the following contributions:

- A modular architecture-agnostic pipeline that takes as input unstructured documents and produces both synthetic QA pairs and a QA model trained on them; We show in Section 4.3 that using this synthetic domain-specific data allows for a dramatic improvement on the QA task compared to baseline state-of-the-art models, especially on unanswerable questions.
- A web-based tool that allows annotators to label various aspects of the synthetic data with ease, alongside guidelines to help ensure consistency and quality in their labels.
- We release[1] this annotation tool and its guidelines for general use. While we use and evaluate this pipeline in the domain of business news, the pipeline is sufficiently flexible to be applied to other domains, including potentially being applicable to abstractive QA.

2 Background and Related Work

Grammaticality Models allow for improving the quality of synthetic data and subsequent performance in downstream tasks by better aligning it with real user data. On benchmark datasets, such as the Corpus of Linguistic Acceptability (CoLA, Warstadt et al., 2019) which contains a wide range of examples from published linguistics literature, current state-of-the-art models (Sun et al., 2019) can achieve a Matthew's Correlation Coefficient score (Matthews, 1975) of approximately 0.775 (Wang et al., 2022), exceeding human performance (0.713, Warstadt et al., 2019) in some cases, though this can vary significantly depending on the sentence's syntactic complexity and length (Warstadt and Bowman, 2020).

Synthetic NLP Data Generation Synthetic data

[1] GitHub

Proceedings of the The 2022 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pages 80 - 97
December 7-11, 2022 ©2022 Association for Computational Linguistics

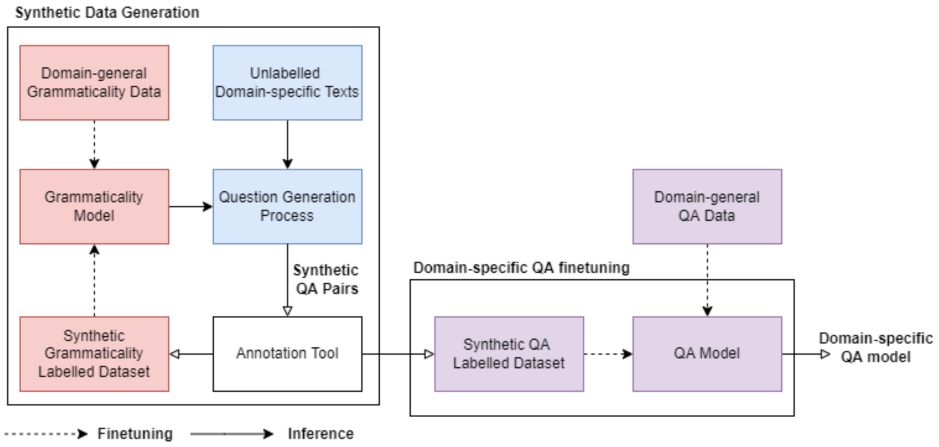

Synthetic Data Generation
Domain-general Grammaticality Data
Unlabelled Domain-specific Texts
Grammaticality Model
Question Generation Process
Synthetic QA Pairs
Synthetic Grammaticality Labelled Dataset
Annotation Tool
Domain-specific QA finetuning
Domain-general QA Data
Synthetic QA Labelled Dataset
QA Model
Domain-specific QA model
Finetuning
Inference

Figure 1: The overall pipeline. The question generation process (blue) generates synthetic QA pairs, which are validated by the grammaticality model. The annotation tool is used to present this data to users for annotation, and the resultant labelled data is then used to fine-tune the grammaticality (red) and QA (purple) models.

generation is an attractive option for dataset creation, especially for domain-specific tasks. Various methods for bootstrapping from smaller datasets have been devised, such as back-translation (Sennrich et al., 2015) and Sibylvariant transformations (Harel-Canada et al., 2022). Backtranslation produces paraphrases through round-trip translation, while Sibylvariant transformations modify or combine texts in predictable ways to create new data with a different label.

Of particular interest are methods that use text generation models to create entirely new data, rather than simply paraphrasing or combining inputs predictably. A variety of these models have been used to generate new QA pairs (Grover et al., 2021), such as the T5 model (Raffel et al., 2020) and BERT (Devlin et al., 2018).

Synthetic data generation can be particularly useful when fine-tuning a model on a specific domain, for which manually-curated datasets may not exist. Whilst high quality datasets such as SQuAD 2.0 (Rajpurkar et al., 2018) do exist for QA tasks, they tend to only have general content, e.g. from Wikipedia. Thus models trained on them often struggle on more domain-specific tasks (Ramponi and Plank, 2020, see also Section 4.3 below).

Evaluation of Synthetic QA Pairs Evaluating Question Generation (QG) models can be difficult due to the nature of the problem: A good question tends to have various qualities (grammatical, answerable, non-trivial to answer, etc.) that are difficult to capture in a single metric, especially one that correlates well with human judgements (Hosking and Riedel, 2019). Nonetheless, several metrics such as BLEU (Papineni et al., 2002) and BERTScore (Zhang et al., 2020) have been proposed, though they rely on having reference questions available and often do not capture whether or not the question is *answerable* (Nema and Khapra, 2018). However, Rajpurkar et al. (2018) show that the use of unanswerable questions when training QA models is important for real-world performance, making it a metric of interest.

Round-trip evaluation, such as the methods proposed by Alberti et al. (2019), allows for evaluating the generated data by checking how consistent downstream model results are when synthetic data is used as the model input, e.g. if the generated answer is found for a synthetic question when the question is input to a QA model. We adopt this approach and discuss it further in Section 4.2.

3 System Overview

Figure 1 shows an overview of our system for creating domain-specific synthetic QA pairs which are used to train downstream models. The QG process (see Section 3.2 for details) creates domain-specific QA pairs from unlabelled texts. This data is then annotated for grammaticality and correctness using the annotation tool, allowing for the creation of two new domain-specific datasets to fine-tune both grammaticality and QA models.

We take a subset of a proprietary knowledge base as our set of input documents and use this to create our domain-specific QA dataset (which we call “SYFTER”). The knowledge base contains

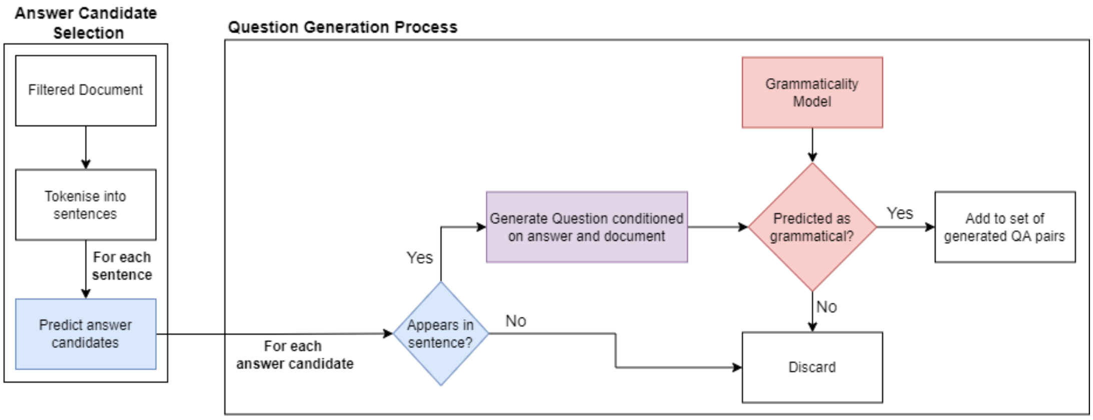

Answer Candidate Selection
Filtered Document
Tokenise into sentences
For each sentence
Predict answer candidates
Question Generation Process
For each answer candidate
Appears in sentence?
Yes
No
Generate Question conditioned on answer and document
Grammaticality Model
Predicted as grammatical?
Yes
No
Add to set of generated QA pairs
Discard

Figure 2: The question generation pipeline.

documents obtained by scraping online articles and is focused on business news, such as information about corporate structures, and is thus quite distinct in subject matter from our external domain-general data (SQuAD 2.0, see Section 3.2).

3.1 Grammaticality Validation

We use a pre-trained BERT model[2] (Devlin et al., 2018) to evaluate the grammaticality of each synthetic question and answer and we discard ungrammatical ones under the intuition that encouraging the synthetic data to be grammatically correct results in the final dataset being more similar to questions posed by real users and improved performance on the downstream task.

We use the "in-domain" data from the Corpus of Linguistic Acceptability (CoLA, Warstadt et al., 2019) dataset to train our grammaticality model in the domain-general setting.

While from a linguistic perspective (Lau et al., 2017), grammaticality can be seen as either a binary or a gradient feature, we use it as a binary label to better standardise with other papers and with CoLA. Furthermore, annotators are unlikely to hold consistent beliefs about the *degree* to which something is ungrammatical, given the high level of subjectivity inherent in such a judgement, and so treating it as binary reduces the potential for noise in the labels.

Because both the CoLA and SYFTER grammaticality datasets have a large degree of class imbalance[3], we use SMOTE (Chawla et al., 2002) to oversample the ungrammatical instances and achieve a uniform class distribution.

3.2 Synthetic Question-Answer Pair Generation

The Question Generation process takes as input a natural language document (in our case, a paragraph or a single sentence) and outputs a QA pair that can be answered from this document. This is done using two models: One to select answer candidates from the document, and one that generates a question based on both the answer and the full document, for each candidate. The full process is shown in Figure 2.

We extend Patil Suraj's question-generation library (Patil, 2022) to work with any SQuAD 2.0-format dataset rather than only ones available from HuggingFace, as well as enabling it to gracefully discard invalid answers without breaking, and partially integrating it into our own pipeline.

We use two separate T5 (Raffel et al., 2020) models fine-tuned on SQuAD V1[4] data for both answer selection and question generation[5], and specify the task at inference time in natural language following the prompting paradigm (Brown et al., 2020). We "highlight" the answer token during question generation as described in (Chan and Fan, 2019).[6] Because the underlying model is abstractive rather than extractive, it occasionally produces answer candidates that do not appear in the context and are thus unusable for extractive QA, which we discard.

Prior to answer selection, we filter out unsuitable input documents in two stages: We first filter out documents that are very short[7] or which match at

[2] bert-base-uncased

[3] Approximately 25% and 10% ungrammatical respectively

[4] Due to time constraints, we did not re-train on SQuAD 2.0, but the model performs well nonetheless (Section 4.2)

[5] valhalla/t5-small-qa-qg-hl and valhalla/t5-base-qg-hl respectively.

[6] E.g. "generate question: The <hl>dog<hl> is red".

[7] Less than 10 tokens

least one of a set of RegEx filters (see Appendix A for details), allowing us to remove any that are clearly semantically null. We then apply a second filter using a BERT Part-of-Speech model[8] such that only documents that contain a verb, or an auxiliary verb and a proper noun, are included so as to remove documents that do not present information that questions can be built around.

Each sentence in each filtered document is input to the answer selection model, which identifies answer candidates within them. Intuitively, a span is an answer candidate if a question can be built around it, and so the model tends to select ones representing entities or relations.

Questions are then generated, conditioned on each answer and the entire associated document, and if validated by the grammaticality model they are added to the synthetic QA dataset.

The resultant dataset can then be input directly into the annotation tool.

An ablation test over the filters (including the grammaticality model) can be found in Appendix C.

3.3 Question Answering

We use an ALBERT (Lan et al., 2019) Question Answering model to predict an answer represented as a span within the document, indicated by two token indices (start and end).

The model is able to provide "null answers", indicating that the question cannot be answered, either directly or by having its prediction changed to the null answer if the null-answer's confidence score is above a "null-answer threshold" (regardless of the original prediction's confidence score).

We utilise SQuAD 2.0 (Rajpurkar et al., 2018) for the initial fine-tuning of our QA model, as it is a large high-quality dataset containing both answerable and unanswerable questions, and as a general-domain dataset it allows us to demonstrate the utility of our domain transfer methods.

The resultant QA model is then fine-tuned on our domain-specific "SYFTER" dataset in order to adapt it to our desired domain, which focuses on news articles about commercial events such as product launches and earnings reports (whereas SQuAD's data comes from Wikipedia and focuses more on history, politics, and geography).[9]

3.3.1 Detecting Unanswerable Questions

During development, we noticed that when trained on a single domain (SQuAD or SYFTER), the QA models could learn to effectively identify if a question from that domain could be answered or not, but performance on this task would drop significantly when trained on both domains.

This was likely due to a combination of our "unanswerable question" label being applied more broadly (to nonsensical questions as well as unanswerable ones), and due to the significant amount of class imbalance in the dataset (especially for the SYFTER data), as well as a small amount of noise in the labels detected through manual inspection.

We explored various methods to resolve this problem when using combined training data, and discuss an ablation study over them in Appendix B, with results in Table 7.

- We appended "source markers" to the end of each question, prior to tokenisation, which indicated the domain that the question came from: either "[SQuAD]" or "[SYFTER]", in order to allow the model to better learn domain-specific features.
- We tuned the 'null-answer threshold" on the validation set.
- We investigated training the model simultaneously for the tasks of both QA and sequence classification as "answerable" / "unanswerable". This follows findings from Crawshaw (2020) that multitask learning can often improve performance, and given the interdependence between question answering and detecting if a question *can* be answered.
- Finally, we used alpha-weighted Focal Loss (Lin et al., 2017) rather than Cross Entropy Loss for sequence classification in the multitask setting to better handle class imbalance.

3.4 Data Annotation

In order to label the synthetic data for supervised training, we created an annotation tool[10] using Streamlit (Treuille et al., 2018) which allows annotators to view model-generated QA pairs, along with their associated context document, and annotate them in various ways. An example of how QA pairs are presented within the tool can be found in Figure 5 in Appendix D.

We used a series of three preliminary studies to

[8] vblagoje/bert-english-uncased-finetuned-pos

[9] SQuAD's domains can be explored here.

[10] A video demo can be found here

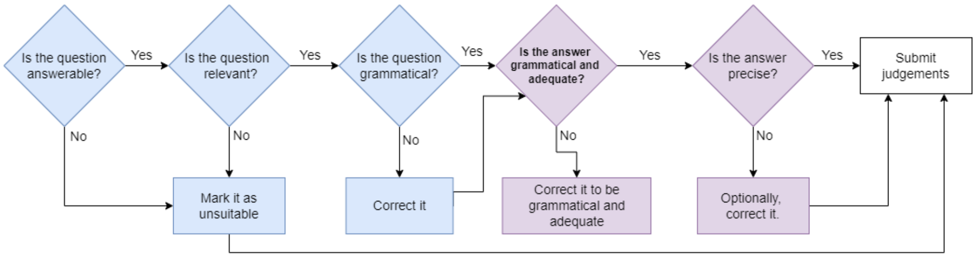

Is the question answerable?
Yes
Is the question relevant?
Yes
Is the question grammatical?
Yes
Is the answer grammatical and adequate?
Yes
Is the answer precise?
Yes
Submit judgements
No
No
No
No
No
Mark it as unsuitable
Correct it
Correct it to be grammatical and adequate
Optionally, correct it.

Figure 3: The annotation process. The answerability and relevance of questions (blue) is dependent on the document, without considering external knowledge. Answers (purple) must appear within the document to be accepted.

Model	Training Data	# Train Examples	Macro F1 Score
BERT	CoLA	10584	61.18
BERT	SYFTER	2796	**75.74**
BERT	CoLA + SYFTER	13608	74.68

Table 1: The Grammaticality model results. The best setting is indicated in **bold** text. “# Train Examples” refers to the data *after* oversampling.

iteratively refine our annotation tool and guidelines, with each study involving 10 participants (who did not participate in subsequent studies). This allowed us to identify and fix any points of misunderstanding before using the tool for the final annotation study on the entire dataset. As with the final annotation study, these were done via Prolific[11] and under the same annotator filters (as well as filtering out previous participants).

Following each preliminary study, we followed up with annotators in cases where they had made unintuitive judgements or appeared to have misunderstood, and used these discussions to refine the guidelines presented. The final guidelines are shown in Appendix D.1.

Each annotator was assigned to a group with two others, and each group of three annotators provided annotations for 2% of the total dataset, with gold labels coming from majority judgements.

The annotation process is shown in Figure 3. Questions marked as unsuitable (for either reason) are not labelled further, and comprise the set of unanswerable questions for the SYFTER domain.

Questions were judged on suitability (whether the question is answerable and relevant to the document) as well as grammaticality.

Grammaticality for both questions and answers was posed to annotators as a question of “reading naturally”, in order to better mimic real user questions and avoid the subjective issues inherent to judging grammaticality.

Answers were judged on both naturalness and quality. In the latter case, an answer was considered “adequate” if it answered the question but had either extraneous details or was missing details, and “precise and correct” if it answered the question with all of the relevant details, but no more.

We asked annotators to rewrite questions and answers that did not read naturally, as well as inadequate answers, and did not allow for the submission of the labels until the texts were corrected or the question was marked as unsuitable (e.g. if they could not be corrected within our constraints).

4 Experiments and Results

The Grammaticality and Question Answering models are tested in both the setting of interest (combined domain-general and domain-specific data) as well as two baseline data settings (domain-general data only[12], and domain-specific data only). This allows us to both measure how useful the synthetic data is as an addition to domain-general data and to also evaluate the feasibility of fine-tuning using *only* synthetic data, which would reduce time and expense significantly given its small size.

The combined test sets for the Grammaticality and QA models are produced by combining the appropriate domain-general data (CoLA or SQuAD) with the domain-specific SYFTER data and then testing the model on this combination dataset.

We evaluate the Question Generation process

[11] https://www.prolific.co/

[12] CoLA for the grammaticality task, SQuAD for the QA task

Test Dataset	QA Model	Exact Match	Similarity
SQuAD 2.0	RoBERTa	**67.81%**	**81.89%**
SYFTER	RoBERTa	64.55%	77.27%

Table 2: Roundtrip evaluation of our QA datasets' quality, using an off-the-shelf QA model. The RoBERTa model was trained on SQuAD 2.0. Best results indicated in bold text.

Document	Question	Answer
"International law firm Ashurst announces the appointment of Matthias Weissinger as partner in Munich.	Who is the new partner of Ashurst in Munich?	Matthias Weissinger
To date we've delivered more than one billion pieces of protective equipment to the frontline.	How many pieces of protective equipment have been delivered to the frontline?	more than one billion
As a major food sector player, Bel fully assumes its duty to do everything possible to ensure the continuity of its operations.	What sector is Bel a major player in?	food

Table 3: Example Question-Answer Pairs Generated from Documents

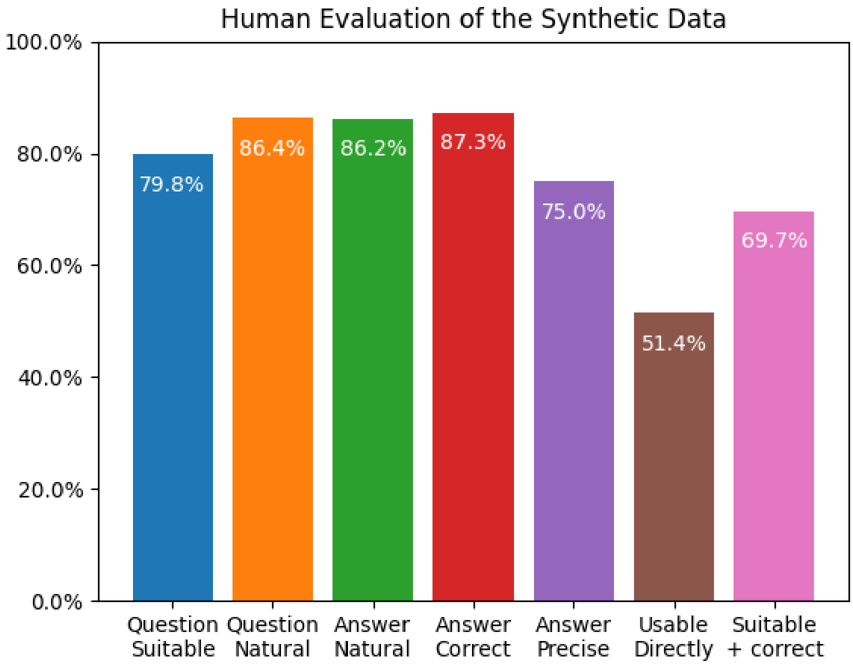


Figure 4: Human Evaluation results on the annotated data. Only QA pairs that had a suitable question were judged further on the other metrics. Percentages shown are based on annotator consensus rather than individual judgements.

Model	Training Data	% Synthetic	Answerable		Unanswerable	Overall	
		Training Data	EM	F1	EM	EM	F1
ALBERT	SQuAD 2.0	0%	**84.87**	**91.09**	12.16	61.06	65.25
ALBERT	SYFTER	100%	53.26	59.71	**72.00**	57.26	63.34
ALBERT	SQuAD 2.0 + SYFTER	0.62%	71.74	83.24	40.00	**64.96**	**74.00**

Table 4: Question Answering model results on the SYFTER test set. The best settings are shown in **bold**.

in both the domain-general and domain-specific settings, but do *not* evaluate the combined setting due to the nature of the evaluation (see Section 4.2).

4.1 Grammaticality Classification

We evaluate the grammaticality model using the model's F1 score, treating grammaticality as a binary sequence classification task, and achieve strong results in both the synthetic-only and combined data settings, as shown in Table 1. The domain-specific model actually performs better than both the domain-general model and the combined-data setting, despite training on only a small amount of synthetic data, indicating the importance of using domain-specific data during training.

4.2 Synthetic Question-Answer Pair Generation

We evaluate the synthetic questions through roundtrip evaluation as discussed in Section 2. For each generated QA pair, we use an off-the-shelf QA model[13] to answer the generated question (based on its associated context) and then compare the answers in two ways: Exact match; and comparing their similarity with their most-similar question at the token level using length-normalised Levenshtein distance (Levenshtein, 1966) via NLTK (Bird et al., 2009). Intuitively, if the question is well-formed and precise, and the answer is relevant to it, the QA model should find the correct answer.

As shown in Table 2, the synthetic data is of high quality, reaching similar levels to SQuAD 2.0, which was manually created by humans. Furthermore, Table 3 shows examples of the synthetic data produced and used. The generated questions are both fluent and of interest, and the answers are both precise and correct. The first question is slightly stilted, but still easily understandable.

Finally, the annotation process can also be thought of as a form of human evaluation and, as shown by Figure 4, the vast majority of the data was found to be of high-quality (suitable, reading naturally, and correct+precise answers). However, 48.6% of the data, including unsuitable questions, did require some input from annotations in some form (not counting data that was imprecise but otherwise good). This indicates that while the data tends to be of high-quality overall, about half of the datapoints do contain a small amount of noise. 69.7% of the questions are suitable and have correct answers, which can be considered the key factors for good synthetic QA data, and as such a high percentage of the data could be used to train a QA system as-is without needing corrections.

4.3 Question Answering

We take approximately 11.6% of the total annotated SYFTER data (117 questions, approximately 21% of which are unanswerable) to use as the QA test set, and split it at the document-level to avoid potential information leaks from the training data.

The QA model is evaluated through both the "Exact Match" (EM) score, and at the token level using F1 score, via the HuggingFace wrapper around the official SQuAD evaluation script. In both cases, the text is first lowercased and normalised to remove articles and standardise whitespace. EM and F1 are identical for unanswerable questions.

We present the results from the best setting, which uses null-answer threshold tuning and multi-task learning *without* Focal Loss (see Appendix B), in Table 4.

The SYFTER-only model performs well despite the SYFTER dataset being much smaller than SQuAD 2.0, and is much better at handling unanswerable questions. By combining the two, we achieve the best overall performance, and maintain reasonable performance on unanswerable questions despite the issues discussed in Section 3.3.1.

5 Conclusion

We present a pipeline for using and evaluating synthetic QA data and an interface for annotating it, as well as annotation guidelines. The combination of domain-general and synthetic data allows our QA model to perform significantly better (+ 9 F1) on domain-specific documents than it did when trained solely on a similar amount of domain-general data. The pipeline is simple to apply to both current and future state-of-the-art models, enabling better performance in low-resource domains.

6 Acknowledgements

This work is supported by an Innovate UK grant under the KTP scheme (KTP 11714) to Rob Procter, Maria Liakata and Filament AI, as well as the Alan Turing Institute (EP/N510129/1). Maria Liakata has been supported by a UKRI/EPSRC Turing AI Fellowship EP/V030302/1.

[13] deepset/roberta-base-squad2, which has strong performance on SQuAD 2 data

7 Limitations

Whilst our system demonstrates that we can achieve significant improvements from synthetic domain-specific data with minimal additional time and expense, it does have certain limitations: We do not consider "adversarial questions" when training, and it thus would likely struggle on these kinds of questions based on findings such as those from Bartolo et al. (2021).

We also found that our synthetic data primarily consists of questions which identify entities (e.g. "Who is the CEO of Microsoft?", "When did Microsoft acquire Bethesda Softworks?", "What are the five principles of good leadership?"), and does not contain many examples of questions about relationships between entities (e.g. "Is selling ice cream more profitable than selling widgets?"), and answers to the latter may be of relatively poor quality.

This is likely due to what appears to be a similar trend in SQuAD V1 that the Question Generation model was trained on: SQuAD primarily asks questions with short entity-focused answers (dates, names, etc.) (Qu et al., 2021) and approximately half of the answers in SquAD (Rajpurkar et al., 2016) are proper nouns, dates, or other numbers indicating that their corresponding questions are likely entity-focused.

The questions of interest to us are generally entity-based and so this limitation does not directly impact our own usage of the model, but we recognise that it potentially limits its applicability to other domains. In the future, the model's performance on non-entity questions could be investigated and improved through tools like AdaTest (Ribeiro and Lundberg, 2022).

The tool also still requires some amount of human involvement to annotate and filter the synthetic data, and the Grammaticality model results (Table 1 indicates that filtering with purely domain-general models would be ineffective. However, it is possible to generate the QA pairs without annotation and, given the high quality of the data (Figure 4), it may be reasonably possible to use the data directly (treating it all as suitable and grammatical) to achieve a still-significant boost to domain-specific performance.

The main problem with not using human annotation would be that our "unanswerable questions" are all ones marked as "unsuitable" by humans, and thus using the synthetic data directly would lead to only having synthetic questions that are considered to be answerable. This could be improved through extending the QG pipeline to also produce deliberately-unanswerable examples, but is not currently possible.

Finally, whilst we use the grammaticality model for validation during the question generation process, we do not train either the Answer Selection or Question Generation models with grammaticality as a second objective function. Training it in a multitask setting would likely have guided it towards producing better input, and may have produced more (valid) data from the corpus.

8 Ethics Statement

Machine learning tasks often involve the potential for ethical issues, especially when using human annotators to label data. We chose to use Prolific[14] as a platform to find and pay annotators, as it offered a reputation for enforcing ethical payments as well as useful filters such as education level and native language.

We also submitted our project to the University of Warwick's internal ethics process, and were approved without having to make any adjustments.

Prolific annotators are paid a fixed amount, but if a task's average hourly payment falls below a minimum (£5 / $6.50 per hour), it is required to rectify this and increase the payments.

The mean rate of pay for annotators was reported as £15.63 during the preliminary studies and £15.50 during the primary annotation study, though these figures are *under-estimates* as our own time-tracking indicates that annotators generally spent a significant amount of time not annotating the data questions (but still recorded by Prolific as being on-task). This is well in excess of the UK living wage of £9.50, as well as the "real living wage" of up to £11.05 proposed by The Living Wage Foundation[15].

The use of synthetic data does have some inherent potential ethical issues: "Model hallucination" is a well-known phenomenon where models can create unfaithful data (e.g. convincing, but false answers to questions) and which can cause real-world harm if the information it provides is acted on (Ji et al., 2022). This can affect our own models if the data generation models hallucinate and lead to the QA model internalising incorrect knowledge.

[14] https://www.prolific.co/

[15] As discussed here.

Thankfully, there are various ways to identify these occurrences and mitigate this harm, including perhaps the simplest method of specifying the context in which the data was created and used at appropriate downstream points, so that users can better assess its veracity for themselves.

To limit this harm, we strongly suggest that other researchers take this into account in their own work, and take the appropriate actions, for instance using human annotators to verify the data and actively designing models to be robust against hallucination, as done in work like Su et al. (2022).

Finally, despite using a model to create our QA data, and the fact that synthetic data can clearly be very useful, bias is still likely to exist in the data (carried forward from both the model's original training data and the human factor of the annotation done), and we suggest that any data produced be investigated and debiased through tools like AdaTest (Ribeiro and Lundberg, 2022).

References

Chris Alberti, Daniel Andor, Emily Pitler, Jacob Devlin, and Michael Collins. 2019. Synthetic qa corpora generation with roundtrip consistency.

Max Bartolo, Tristan Thrush, Robin Jia, Sebastian Riedel, Pontus Stenetorp, and Douwe Kiela. 2021. Improving question answering model robustness with synthetic adversarial data generation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python: analyzing text with the natural language toolkit.* " O'Reilly Media, Inc.".

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared 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 M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners.

Ying-Hong Chan and Yao-Chung Fan. 2019. A recurrent BERT-based model for question generation. In *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*, pages 154–162, Hong Kong, China. Association for Computational Linguistics.

N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. 2002. SMOTE: Synthetic minority oversampling technique. *Journal of Artificial Intelligence Research*, 16:321–357.

Michael Crawshaw. 2020. Multi-task learning with deep neural networks: A survey.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding.

Steven Y. Feng, Varun Gangal, Jason Wei, Sarath Chandar, Soroush Vosoughi, Teruko Mitamura, and Eduard Hovy. 2021. A survey of data augmentation approaches for nlp.

Khushnuma Grover, Katinder Kaur, Kartikey Tiwari, Rupali, and Parteek Kumar. 2021. Deep learning based question generation using t5 transformer. In *Advanced Computing*, pages 243–255, Singapore. Springer Singapore.

Fabrice Harel-Canada, Muhammad Ali Gulzar, Nanyun Peng, and Miryung Kim. 2022. Sibylvariant transformations for robust text classification. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1771–1788, Dublin, Ireland. Association for Computational Linguistics.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Jack W. Rae, Oriol Vinyals, and Laurent Sifre. 2022. Training compute-optimal large language models.

Tom Hosking and Sebastian Riedel. 2019. Evaluating rewards for question generation models.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Yejin Bang, Andrea Madotto, and Pascale Fung. 2022. Survey of hallucination in natural language generation.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations.

Jey Han Lau, Alexander Clark, and Shalom Lappin. 2017. Grammaticality, acceptability, and probability: A probabilistic view of linguistic knowledge. *Cognitive Science*, 41(5):1202–1241.

Vladimir I. Levenshtein. 1966. Binary codes capable of correcting deletions, insertions, and reversals.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. 2017. Focal loss for dense object detection.

B.W. Matthews. 1975. Comparison of the predicted and observed secondary structure of t4 phage lysozyme. *Biochimica et Biophysica Acta (BBA) - Protein Structure*, 405(2):442–451.

Preksha Nema and Mitesh M. Khapra. 2018. Towards a better metric for evaluating question generation systems. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3950–3959, Brussels, Belgium. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, , and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation.

Suraj Patil. 2022. Question Generation using transformers.

Fanyi Qu, Xin Jia, and Yunfang Wu. 2021. Asking questions like educational experts: Automatically generating question-answer pairs on real-world examination data.

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.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for squad.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text.

Alan Ramponi and Barbara Plank. 2020. Neural unsupervised domain adaptation in nlp—a survey.

Marco Tulio Ribeiro and Scott Lundberg. 2022. Adaptive testing and debugging of NLP models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3253–3267, Dublin, Ireland. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Improving neural machine translation models with monolingual data.

Dan Su, Xiaoguang Li, Jindi Zhang, Lifeng Shang, Xin Jiang, Qun Liu, and Pascale Fung. 2022. Read before generate! faithful long form question answering with machine reading.

Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Xuyi Chen, Han Zhang, Xin Tian, Danxiang Zhu, Hao Tian, and Hua Wu. 2019. Ernie: Enhanced representation through knowledge integration.

Adrien Treuille, Thiago Teixeira, and Amanda Kelly. 2018. Streamlit.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2022. Glue benchmark leaderboard.

Alex Warstadt and Samuel R. Bowman. 2020. Linguistic analysis of pretrained sentence encoders with acceptability judgments.

Alex Warstadt, Amanpreet Singh, and Samuel R. Bowman. 2019. Neural network acceptability judgments.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert.

A RegEx Document Filters

Table 5 shows the different RegEx filters that we apply to documents in order to filter out ones that are likely to be difficult to select valid answers from. Documents are filtered if *any* substring in them is a match for the expression.

The first expression, which filters out documents that appear to be too similar to contracts, additionally contains certain *whitelist* expressions which prevent otherwise-matching documents from being removed. These can be seen in Table 6. In order to be whitelisted, the text that matched the initial filter must fully match the whitelist expression (though the entire document does not have to match).

For clarity when dealing with leading/trailing whitespace, each expression is wrapped in "double quotes", but these quotes are not part of the actual expression. Matches with each expression are **emphasised** for clarity.

RegEx Expression	Intended Matches	Example Match
“` ?\([0-9A-Za-z]+\)(\([0-9A-Za-z]+\))*`”	Contract-like documents	“B 1: Financial Instruments according to Regulation 17**(1)(a)** of the Regulations”
“`^[0-9]+\.? ?.+`”	Numeric List	“**1. Reassure customers and employees**”
“`^[ivx]+\.? .+`”	Roman-numeric List	“**xi If the financial instrument has such a period**”
“`\[ ?\]`”	Empty square brackets	“**[ ]** An acquisition or disposal of financial instruments”
“`Regulation(s)? [0-9]+`”	Regulations contract-like	“B 2: Financial Instruments with similar economic effect according to **Regulation 17** of the Regulations”
“`^.{0,15}$`”	Very short documents	“**content**”
“`^(.{0.5})?\(.+\).{0,5}$`”	Mostly in brackets	“**(please tick the appropriate box or boxes):**”

Table 5: RegEx Filters for Documents

RegEx Expression	Purpose	Example Documents Whitelisted
“` ?\([A-Z]+s?\)`”	Allow acronyms	“CPE Lite is Huawei’s latest mini customer premises equipment **(CPE)**.”
“` ?\([A-Z]?[0-9a-z]{4,}\)`”	Allow short bracketed words	“Bel reported strong sales momentum in the first two months of the year in global**(mature)** markets”

Table 6: RegEx Whitelists for Documents, applied to the “Contract-like” filter.

B Question Answering Ablation

We performed an ablation study over the Question Answering Model components discussed in Section 3.3.1, and found that in some cases they significantly improve the performance on unanswerable questions, especially the use of multitask learning. The results of this ablation are shown in Table 7.

Whilst we found that some settings (Source Markers, Focal Loss) did not appear to be useful, we nonetheless believe that the utility of source markers when using more domains would be an interesting avenue for future investigation.

C Question Generation Filter Ablation

We performed an ablation study over the Question Generation filters discussed in Section 3.2 and found that the individual filters tend to have a significant impact on the model's performance on unanswerable questions, but relatively little when considering answerable questions. Given that the filters were primarily designed to filter out documents that were likely to produce low-quality unanswerable questions, this is as expected. The set of filters that we used does not provide the best *overall* F1 Score, but provides a model whose performance is significantly more balanced than the nominally best-performing model, a trait that we found valuable.

For these tests, we trained and tested the QA model *only* on SYFTER data so as to most clearly see the effects of the filter(s) used (since SQuAD data is not filtered in our pipeline).

Source Markers	Threshold Tuning	Multitask	Focal Loss	Performance Gain (F1)		
				Answerable	No Answer	Overall
x	x	x	x	91.02	72.97	85.11
✓	x	x	x	- 2.2	+ 0	- 1.48
x	✓	x	x	- 0.66	+ 1.35	+ 0
x	x	✓	x	**- 1.98**	**+ 4.06**	**+ 0**
x	x	✓	✓	- 2.14	+ 0	- 1.44
✓	✓	✓	✓	- 1.91	+ 1.35	- 0.84

Table 7: Relative performance gains on the ALBERT QA model in different training settings. A checkmark indicates that the component was used, an "x" that it was not. Focal loss is only applicable in the multitask setting. Best setting shown in **bold**.

Filter				Performance Gain (F1)		
Length	RegEx	Part of Speech	Grammaticality	Answerable	No Answer	Overall
x	x	x	x	72.35	40.00	66.22
✓	x	x	x	65.60	48.00	62.16
x	✓	x	x	73.8	24	64
x	x	✓	x	**73.67**	52	**69.5**
x	x	x	✓	71.95	36	65.24
✓	✓	✓	✓	59.71	**72.00**	63.34

Table 8: Relative QA performance gains on the SYFTER test set model using different SYFTER training data filtered in different ways. A checkmark indicates that the component was used, an "x" that it was not. Best setting shown in **bold**. Only SYFTER data was used for training.

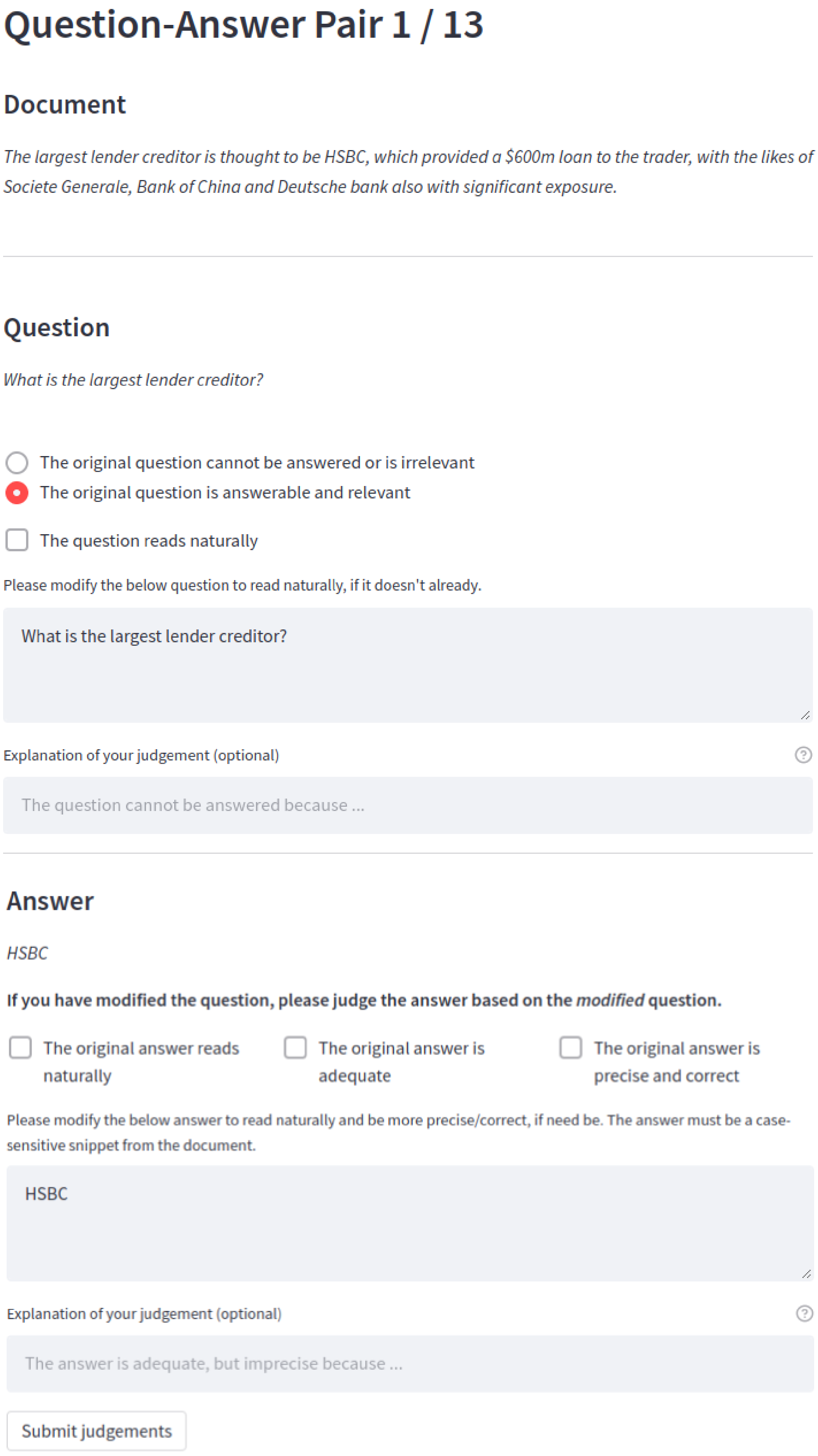
D Annotation Tool

Figure 5 shows an example of how QA Pairs are presented to annotators in the annotation tool. See Section 3.4 for details.

A video demo of the tool can be found here

D.1 Annotation Guidelines

We present a set of annotation guidelines which can be given to annotators in order to obtain consistent labels by “calibrating” their expectations of what is and is not a valid QA pair. The guidelines for labelling questions can be found in Figure 6 and for answers in Figure 7.



Question-Answer Pair 1 / 13

Document

The largest lender creditor is thought to be HSBC, which provided a $600m loan to the trader, with the likes of Societe Generale, Bank of China and Deutsche bank also with significant exposure.

Question

What is the largest lender creditor?

○ The original question cannot be answered or is irrelevant
● The original question is answerable and relevant

□ The question reads naturally

Please modify the below question to read naturally, if it doesn't already.

What is the largest lender creditor?

Explanation of your judgement (optional)

The question cannot be answered because ...

Answer

HSBC

If you have modified the question, please judge the answer based on the *modified* question.

□ The original answer reads naturally
□ The original answer is adequate
□ The original answer is precise and correct

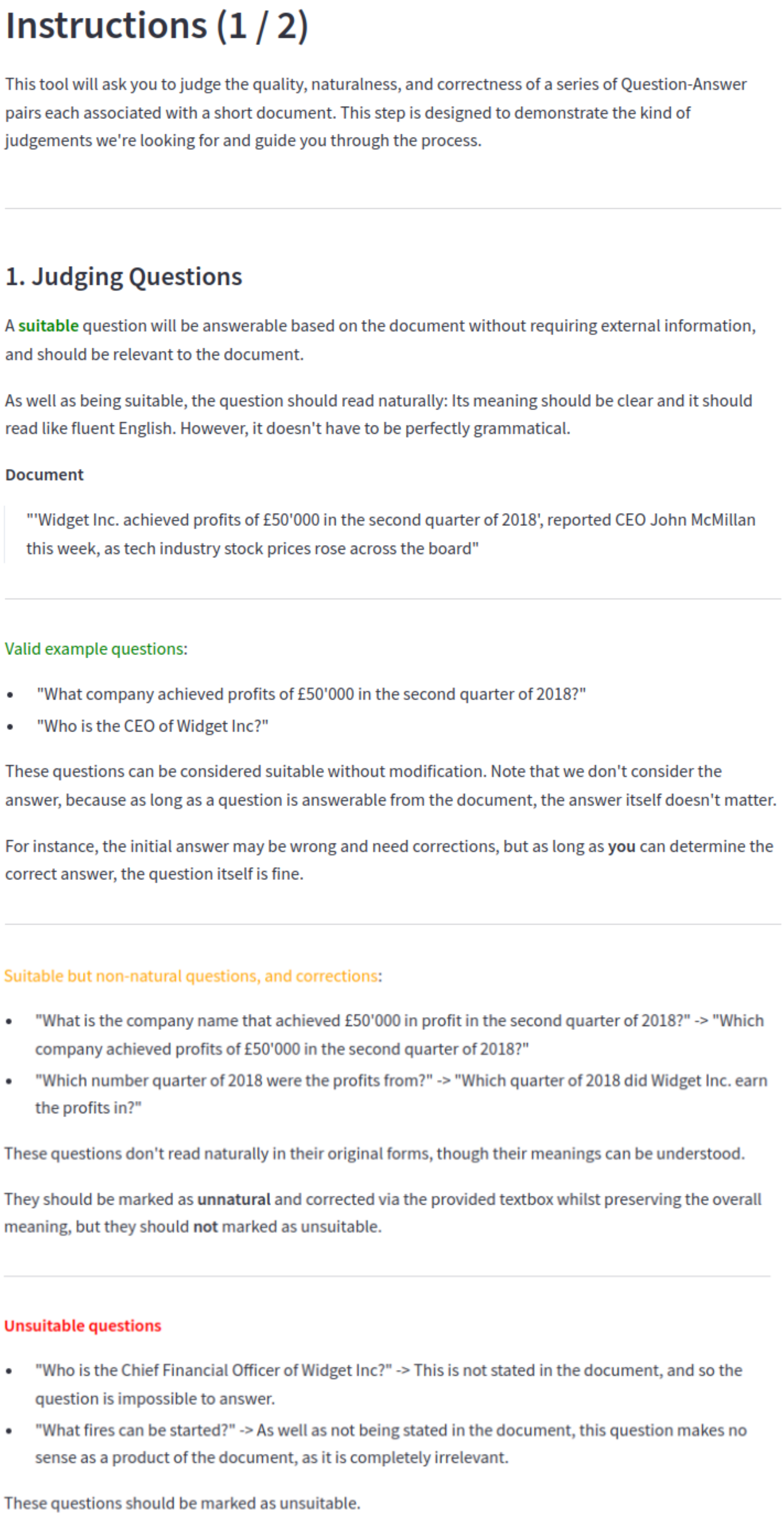
Please modify the below answer to read naturally and be more precise/correct, if need be. The answer must be a case-sensitive snippet from the document.

HSBC

Explanation of your judgement (optional)

The answer is adequate, but imprecise because ...

Figure 5: An example of how QA pairs are presented in the annotation tool.



Instructions (1 / 2)

This tool will ask you to judge the quality, naturalness, and correctness of a series of Question-Answer pairs each associated with a short document. This step is designed to demonstrate the kind of judgements we're looking for and guide you through the process.

1. Judging Questions

A **suitable** question will be answerable based on the document without requiring external information, and should be relevant to the document.

As well as being suitable, the question should read naturally: Its meaning should be clear and it should read like fluent English. However, it doesn't have to be perfectly grammatical.

Document

> "'Widget Inc. achieved profits of £50'000 in the second quarter of 2018', reported CEO John McMillan this week, as tech industry stock prices rose across the board"

Valid example questions:

- "What company achieved profits of £50'000 in the second quarter of 2018?"
- "Who is the CEO of Widget Inc?"

These questions can be considered suitable without modification. Note that we don't consider the answer, because as long as a question is answerable from the document, the answer itself doesn't matter.

For instance, the initial answer may be wrong and need corrections, but as long as **you** can determine the correct answer, the question itself is fine.

Suitable but non-natural questions, and corrections:

- "What is the company name that achieved £50'000 in profit in the second quarter of 2018?" -> "Which company achieved profits of £50'000 in the second quarter of 2018?"
- "Which number quarter of 2018 were the profits from?" -> "Which quarter of 2018 did Widget Inc. earn the profits in?"

These questions don't read naturally in their original forms, though their meanings can be understood.

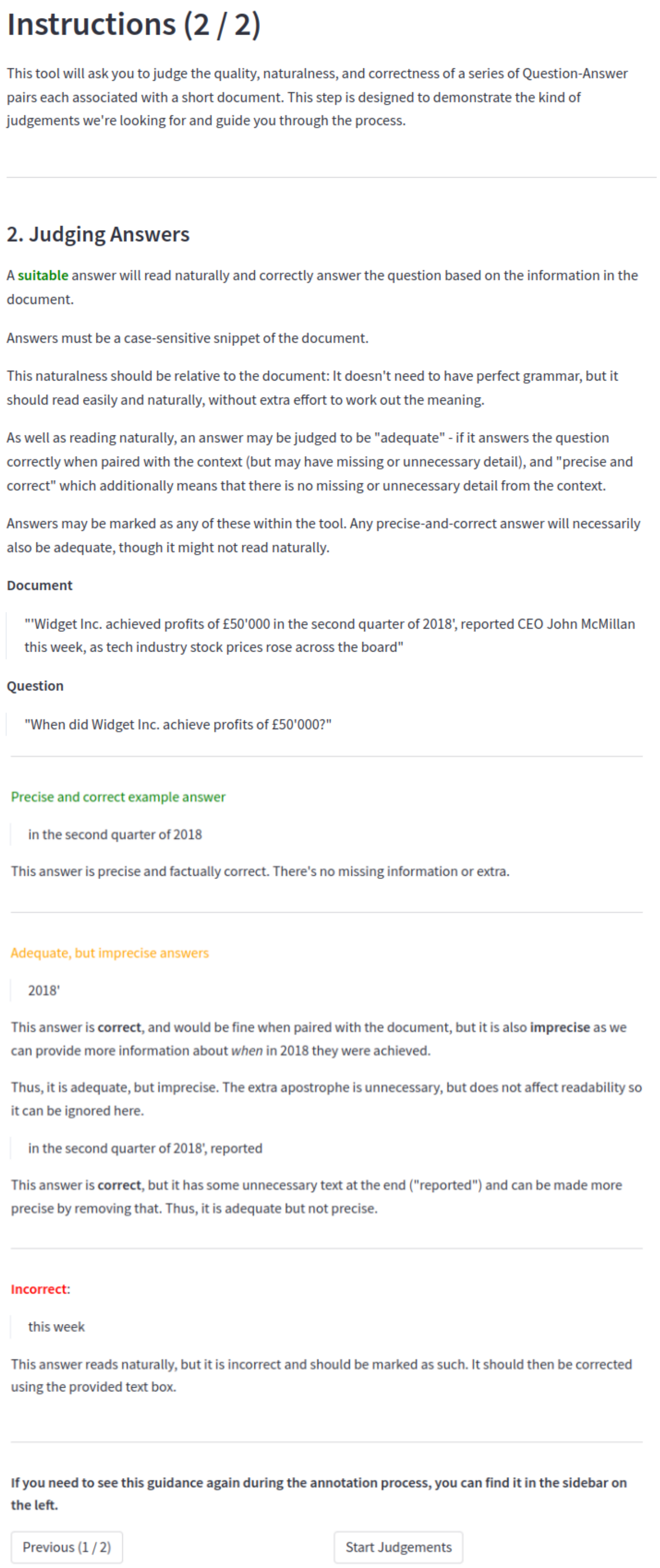
They should be marked as **unnatural** and corrected via the provided textbox whilst preserving the overall meaning, but they should **not** marked as unsuitable.

Unsuitable questions

- "Who is the Chief Financial Officer of Widget Inc?" -> This is not stated in the document, and so the question is impossible to answer.
- "What fires can be started?" -> As well as not being stated in the document, this question makes no sense as a product of the document, as it is completely irrelevant.

These questions should be marked as unsuitable.

Figure 6: Annotation guidelines for judging question suitability and naturalness.



Instructions (2 / 2)

This tool will ask you to judge the quality, naturalness, and correctness of a series of Question-Answer pairs each associated with a short document. This step is designed to demonstrate the kind of judgements we're looking for and guide you through the process.

2. Judging Answers

A **suitable** answer will read naturally and correctly answer the question based on the information in the document.

Answers must be a case-sensitive snippet of the document.

This naturalness should be relative to the document: It doesn't need to have perfect grammar, but it should read easily and naturally, without extra effort to work out the meaning.

As well as reading naturally, an answer may be judged to be "adequate" - if it answers the question correctly when paired with the context (but may have missing or unnecessary detail), and "precise and correct" which additionally means that there is no missing or unnecessary detail from the context.

Answers may be marked as any of these within the tool. Any precise-and-correct answer will necessarily also be adequate, though it might not read naturally.

Document

> "'Widget Inc. achieved profits of £50'000 in the second quarter of 2018', reported CEO John McMillan this week, as tech industry stock prices rose across the board"

Question

> "When did Widget Inc. achieve profits of £50'000?"

Precise and correct example answer

> in the second quarter of 2018

This answer is precise and factually correct. There's no missing information or extra.

Adequate, but imprecise answers

> 2018'

This answer is **correct**, and would be fine when paired with the document, but it is also **imprecise** as we can provide more information about *when* in 2018 they were achieved.

Thus, it is adequate, but imprecise. The extra apostrophe is unnecessary, but does not affect readability so it can be ignored here.

> in the second quarter of 2018', reported

This answer is **correct**, but it has some unnecessary text at the end ("reported") and can be made more precise by removing that. Thus, it is adequate but not precise.

Incorrect:

> this week

This answer reads naturally, but it is incorrect and should be marked as such. It should then be corrected using the provided text box.

If you need to see this guidance again during the annotation process, you can find it in the sidebar on the left.

Figure 7: Annotation guidelines for judging answer naturalness and quality.