You Make me Feel like a Natural Question: Training QA Systems on Transformed Trivia Questions

Anonymous ACL submission

Abstract

Training question answering (QA) and information retrieval systems for web queries requires large, expensive datasets that are difficult to annotate and time-consuming to gather. Moreover, while "natural" datasets of information-seeking questions are often prone to ambiguity or ill-formed, for many languages there are troves of freely available carefully crafted questions. Thus, we automatically generate shorter, information-seeking questions, resembling web queries in the style of the Natural Questions (NQ) dataset from longer trivia data. However, because not all of the generated questions are high quality or match the desired domain, we also use a classifier trained on linguistic, grammatical, style, and topic dependent feature to find questions that match traditional training data in style and topic. Training a QA system on these transformed questions is a viable strategy for alternate to more expensive training setups and contrast the final systems.

1 Introduction

Question answering is a central problem in AI research. One way of understanding *why* people ask question was was explained in Rogers et al. (2023): questions come from either an information seeking paradigm (Voorhees, 2019, henceforth information-seeking) or a probing, evaluative paradigm (Turing, 1950, probing). While it is easy to get *questions* in the information-seeking paradigm, because the asker by definition does not know the answer, additional annotation to find the answer is expensive.

Moreover, Boyd-Graber and Börschinger (2020) argue that probing questions are fundamentally better because they have processes to avoid ambiguity (Min et al., 2020), false presuppositions (Yu et al., 2022), and are more artfully crafted.

However, these bold claims have not been supported by hard evidence. The dataset from Kwiatkowski et al. (2019) is more expensive than their probing counterparts, which are mostly written by trivia enthusiasts. While large corporations can gather many "natural" information-seeking questions "for free", these questions critically do not include the correct *answers*.

This paper investigates whether we can transform the unrealistic sentences harvested from trivia community into questions that resemble natural questions. Such a process, rather than requiring expensive annotations can be done with rule-based transformations (Section 3). We then select the most natural questions from those transformed questions using a classifier to create a QA system to evaluate on real NQ test set.

We consider two experimental settings: zero-shot and supervised. The zero-shot setting imagines a world without NQ: can we build a system that does similarly well as existing systems with our transformed probing questions? In some ways, this is an unfair comparison, as Section 4 still evaluates on NQ data. In our other experimental setting (supervised, Section 5) we augment the NQ data with our transformed questions to improve algorithms that have been trained on NQ.

Our experiments demonstrate that QB questions could replace the questions in NQ dataset in the zero-shot setting and supplement them in supervised QA systems (Section 7).

2 An Artful but Arcane Trivia Dataset

Consider what you might be asked in the quizbowl (QB) format (Boyd-Graber et al., 2012):

> A radio mast named for this city was the world's tallest structure until the mast collapsed in 1991. This capital contains a skyscraper formerly known as the Joseph Stalin Palace of Culture and Science. A landmark called Sigismund's Column commemorates Sigismund III Vasa, who moved his capital from Kraków to this city on the Vistula River. A 1943 Jewish ghetto uprising occurred in—for 10 points—what Polish capital?

First, for text like this where its goal is to elicit information, we define it an *elicitation*; our goal is

transforming these elicitations into grammatically "real" and plausible "natural" questions.

Second, these elicitations are longer and complex than other QA datasets.[1] This is because in QB dataset clues are introduced pyramidally which means harded, more obscure information comes first (Rodriguez et al., 2021). For example, "moved his capital from Kraków to this city on the Vistula" requires ability to decide not just what to answer, but also *when* to answer (He et al., 2016).

Our goal is to avoid this baroque complexity and use what is actually useful in the QB format: a series of clues reveal information that an expert author thought was noteworthy about *Warsaw*: key sites that commemorate its history, rulers who made it the capital, and what country it's a capital of. As each of these could become a standalone question, our goal is to turn each clue in into a question similar to Natural Question (Kwiatkowski et al., 2019), a dataset collected by Google from questions people asked online. These questions are substantially shorter, typically only a handful of words, and have an answer annotated from a Wikipedia page.

2.1 Comparison with NQ Datasets

The released QB and NQ datasets seem comparable (QB: $800k$ elicitation and answer samples and NQ: 58860 samples); however, there exists substantial difference in cost, quality, and quantity. Each original QB elicitation generates on a average of seven QB question. The average sentence length for each elicitation is 12 words. In NQ, the average sentence length is eight words (Kwiatkowski et al., 2019). The NQ questions were composed based on unique heuristics.[2]

First, while QB elicitations are unambiguously paired with the answer by the author, NQ questions must be laboriously annotated by paid workers. While Google has not officially released costs, the convoluted, painstaking process and the lack of reproduction since 2019 suggests that it wasn't cheap. QB, on the other hand, is a byproduct of trivia enthusiast communities who release their old questions into the public domain. From the QA researcher's perspective, the elicitations are at free cost.

[1]That is because it is designed to be interrupted as it is read out loud: it is a sequence of many facts about Warsaw going from obscure to well-known: whoever knows the most about Warsaw should be able to answer the question sooner.

[2]For example, the questions start with "who", "when" or "where" followed by a finite form of "do" or a modalverb (Kwiatkowski et al., 2019)

The process of constructing this dataset also points to quality considerations. Because the author knows the answer during writing and specifically wants to discourage ambiguity (Boyd-Graber and Börschinger, 2020), they will avoid the ambiguity (Min et al., 2020) and false presuppositions (Kim et al., 2021) that are often in NQ. If we can faithfully extract these artfully-crafted clues from QB questions, these questions may be of higher quality than NQ questions.

Finally, because each QB elicitation contains many clues, the potential size of a transformed dataset could be fivefold larger than NQ. And while the NQ dataset may only ask a single question about a rare entity, this is never the case for QB: a single original elicitation would produce several clues about an entity, allowing a model to understand more about each potential answer.

3 Transforming into a Natural Question

As mentioned in the previous section, we can obtain many questions from successfully converting QB elicitations into NQ-like questions. Having motivated why we want to convert QB elicitations to NQ questions, this section outlines our method of converting the long QB elicitations into multiple relevant NQ-like questions (Figure 1).

3.1 Generating Candidates

Many of the transformations we describe in this section depend on an initial syntactic analysis. First, we create a dependency parse (Nivre, 2010) of the sentence. Moreover, some parts of the elicitations do not resemble how questions are asked. For example, many of the questions are statements of fact about the target entity "she was the last Queen of Hawaii" or "this element is mined from bauxite". To transform these mentions into something that looks like a question, we find mentions that are coreferent with the answer.

Conjunction and Removing Clauses Given these candidates, we then need to extract the minimal facts that would form the basis of a question. For example, if the QB elicitation had "he wrote *Animal Farm* and 1984", this can become two facts: "he wrote *Animal Farm*" and "he wrote *1984*". Thus, we construct independent clauses by extracting spans that contain the mention ("he"), a verb ("wrote"), and one member of a conjunction (either of the two works). Similarly, we can

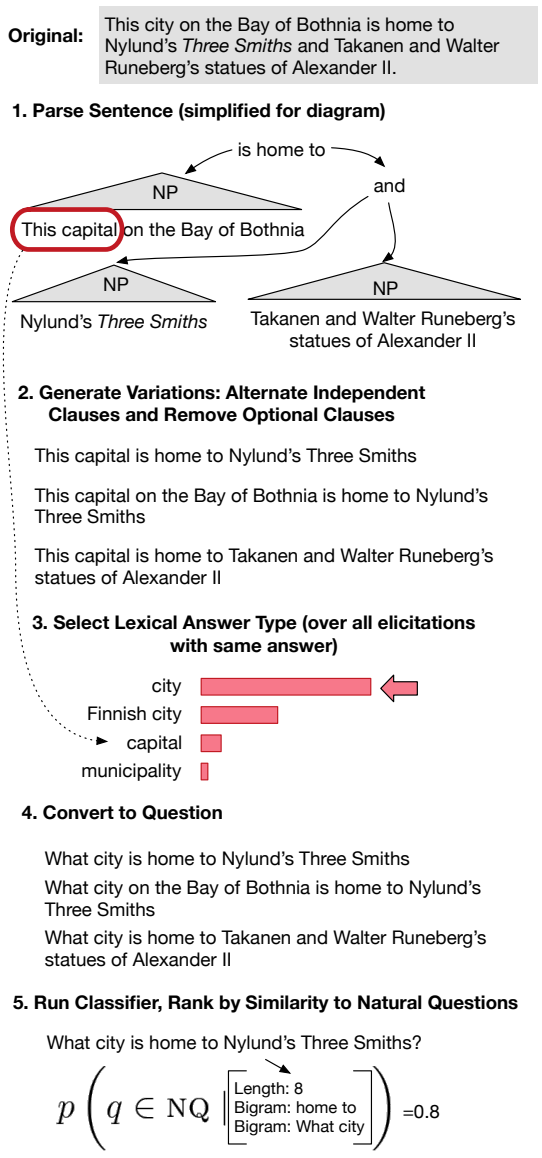

Figure 1: In the process of creating information-seeking style questions from probing elicitations, (1) we take each sentence from the paragraph-long elicitations, and parse it. (2-3) The parsed sentences are transformed into variants, (4) that are finally turned into information-seeking questions. (5) We then use a classifier to detect the most resembling NQ question.

sometimes remove clauses: "this author who graduated Eton College wrote *Homage to Catalonia*" can be simplified to "this author wrote *Homage to Catalonia*".

Cannonical Answer Type Next, we need to figure out what kind of answer the question is looking for. This is important because sometimes questions written in QB's pyramidal style use oblique references particularly at the beginning of the question: "substance" for zinc, "creator" for Chinua Achebe, or "polity" for Bangladesh. However, these are rarer than the most straightforward and direct references. For example, zinc is most often asked about using "what element", Chinua Achebe with "what playwright", and Bangladesh with "what nation". Thus, we group all QB elicitations that have the same answer and for each answer find the most frequent string used to ask about the answer. These cannonical answer types then replace the mentions in the original question.

Imperative to Interrogative The most obvious difference between QB elicitations and NQ questions is that QB elicitations are not grammatical questions: rather, they are declarative statements about the answer (hence why we are going through the trouble of calling them elicitations). Or (often in the last sentence) an imperative statement like "name this first prime minister of Canada"; because these lack a mention, we generate a synthetic mention that makes the object of the imperative verb the question: "who was the first prime minister of Canada" by mapping the cannonical answer type to its WordNet (Fellbaum, 1998) hypernym and applying the appropriate question word (e.g., `person.n.01` maps to "who", `time_period.n.01` maps to "when"). For example, "he wrote Animal Farm" becomes candidates "who wrote Animal Farm."

Additional Heuristics Through observation of the linguistic and grammatical style of NQ we add additional heuristics to further improve the candidates such as **removing punctuation** and **adding subject** (full list in Appendix A).

3.2 Selecting Candidates

The process outlined above will result in many questions that insufficiently resemble the information-seeking questions we want to emulate: some are too short or long, do not make sense, or still look too much like a probing QB elicitations. Like how Goodfellow et al. (2014) use a classifier to filter the outputs of an automatic generative process, we identify the best examples from the above process. We use a simple logistic regression classifier[3] (Cox, 1958) trained on the generated NQ-like

[3]In the introduction, we argued that our approach was cheaper than NQ. At first glance, using the NQ dataset to train this classifier seems to contradict the argument. However, while we are using NQ *questions*, we are critically not using the answers to the questions, which (unlike the answers) are

examples (through the process described in the previous section) as negative examples and with real NQ examples as positive examples.

Nonetheless, our features identify question topics and formats that occur frequently in NQ. For example, the bigram "who played", reflects NQ's emphasis on popular culture; starting questions with "how", "when", or "where" recapitulates the process for harvesting NQ; and short questions have the highest feature weight, emphasizing that NQ questions are short (Table 5).

3.3 LLM Conversion Baseline: Llama 2

As a baseline, we convert QB elicitations into questions through prompting LLAMA2 (Touvron et al., 2023), a generative text model.[4] For fair comparison, we separate clues from the QB elicitations; then feed them to LLAMA2 and ask it to produce a natural question. As in the above pipeline, we identify the lexical answer type (e.g., "this person") and ask LLAMA2 to formulate a query[5] that could be used as a Google search. Afterward, we use the same classifier to select examples from the LLM baseline.

4 Training a zero-shot QA System with Synthetic Data

This section trains systems that do not use NQ data. We call this setting zero-shot, where a question q is given to the model as the input. Based on that input, the model generates the answer a denoted by $p(a|q, \theta)$ where θ is the model .

4.1 Challenges in zero-shot QA System

However, there are some challenges in the design with the zero-shot QA system. Firstly, some state-of-the-art zero-shot systems use NQ data in training (e.g., finetuning or tuning model parameters). For example, Sun et al. (2023) uses NQ training data to fine-tune their retriever component with the NQ train set when testing on NQ test set. Therefore, although these systems are claiming to be zero-shot, the usage of NQ train set impacts the score.

Secondly, these models use large language models such as GPT (Brown et al., 2020) or InstructGPT (Ouyang et al., 2022) in their pipeline, which are not disclosed completely their training data (Shi et al., 2023a). This lack of information poses serious challenges in the zero-shot evaluation (Oscar Sainz, 2023; Narayanan, 2023; Magar and Schwartz, 2022).

Thus, to validate the zero-shot property, we probe GPT to see if it is aware of NQ answers (Table 1). As it may be aware of some answers by coincidence, we focus on *wrong* NQ answers (manually detected). This is the clearest signal that the model has seen the NQ data's answers, as annotation errors are less likely to be by coincidence. We also probe for time-sensitive questions.

Therefore, in our zero-shot QA systems, we are not using any large language models. We are using the heuristics and classifiers to develop our dataset and in our model selection, we have ensured that the models are not pretrained or finetuned on NQ dataset.

4.2 Training

Having described how to generate question–answer pairs that resemble information-seeking paradigm questions, we now want to see how useful they are for training a traditional QA system; how QB questions can act as a replacement of NQ data in the training process. Our goal is to create a QA system with the same accuracy as the original NQ dataset while training on the QB dataset, so this is an upper bound. For evaluation, we have tested the systems on the NQ test set to validate its performance without training any NQ data.

In the zero-shot setting, the system is trained with our QB questions. In this setting, we have ensured to never use questions of the training split of the NQ dataset.

In the zero-shot setting, we have only used the questions generated from the QB dataset. In this scenario, we have experimented with only the QB_ALONE dataset ensuring no NQ train set pollute the result. We have conducted experiments with only QB_ALONE dataset to see whether training with transformed QB_ALONE questions can achieve comparable performance. We have replaced the training dataset of the state-of-the-art QA systems with our QB_ALONE questions and tested with the NQ test set.

4.3 Zero-shot QA systems

For the above mentioned analysis, we have not used any GPT-based methods in our system. We selected two systems that have shown high accuracy

expensive to collect for NQ.

[4] https://huggingface.co/meta-llama/Llama-2-7b-hf

[5] This is the *QUESTION*. Ask about "this person" in the question. Your question's correct answer should be *ANSWER*. Make sure the answer is not in your question. Make the question as natural as a google search query

NQ question	NQ answer (wrong)	Real answer	GPT given answer
Who sang the most number of songs in the world	Asha Bhosle	Lata Mangeshkar	Asha Bhosle
Who introduced the first christmas tree to the uk	Charlotte of Mecklenburg - Strelitz	Prince Albert, Queen Victoria's consort	Queen Charlotte
Total number of death row inmates in the us	2,718	2,331	Over 2,400 people
Who is next in line to be the monarch of england	Charles , Prince of Wales	Prince William	Charles, Prince of Wales
What age is the oldest living person in the world	117	116 years	117

Table 1: To determine whether NQ is in the training data of GPT, we take the answers given by GPT 3.5. If the answer is the same as given in NQ dataset, we can assume it has seen those dataset.

on traditional NQ training: Deep Passage Retrieval (Karpukhin et al., 2020b, DPR) and Retrieval-Augmented Language Modeling Framework (Shi et al., 2023b, REPLUG) for open-domain question answering. These systems trained from the ground-up in our method. **DPR** (Karpukhin et al., 2020a) extracts the answer from a context which is extracted using passage retriever models. We train DPR on the questions, answers, and context passages for QB dataset and the NQ-like generated questions dataset (ours). In training, we generate the positive context by collecting passages that contain answer string, and negative context otherwise (Example in Appendix 6). In **REPLUG** (Shi et al., 2023b), the retrieval model finds the most appropriate passage from a large corpus; then the model produces more accurate answers by augmenting retrieved information to the input context.

4.4 Training Data

We will be comparing all of our generated datasets with the original NQ dataset (**NQ**). Our goal is to create a QA system with the same accuracy as the original NQ dataset while training on the QB dataset, so this is obviously an upper bound. In this zero-shot experiment, we have used different percentage of QB generated questions for training the model. For example, **QB-Trans-10**, represents ten percent of all of the filtered and transformed data set of QB data selected based on the classifier (Section 3.2).

We compare this traditional training regime with several training sets derived from QB. we use individual elicitation sentences from the QB dataset *without* any transformation: **QB-Raw**. While we expect this to do poorly, it shows how much our transformation improves upon the original dataset.

Next, we compare against all transformed sentences from our syntactic-based method (**QB-Trans-100**) compared to the LLM baseline (**QB-Llama2**). For both of the the transformations, we compare against different sampling approaches: uniformly at random, sorted by classifier, or weighted by classifier.

4.5 Results and Analysis

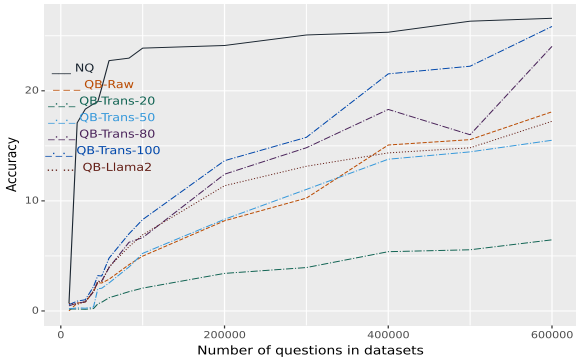

Figure 2: **DPR**: As expected, **QB-Trans-100** without any NQ data comes within 5 points of a model trained on **NQ**. Training on the full QB-Trans and evaluating on it produces the highest accuracy system with DPR. However, the a percentage of that datasets from our systematic conversion (**QB-Trans-80**) reaches a substantial fraction of the accuracy. This does better than conversions created by prompting a LLM.

Our transformations lag behind a model trained directly on NQ by only about three points, while the LLM lags by over ten points. We have seen that our QB_ALONE data can be applied to different QA systems and achieve comparable performance (Figure 2 and 3).

Even the worst transformed questions from the QB dataset are better than many of the questions produced by the LLM. For example, the original QB elicitation has the clue "In one of this man's paintings, one character oddly uses her left hand to grasp the red-cloaked character's chin while her right hand sits at his knee". When we convert it

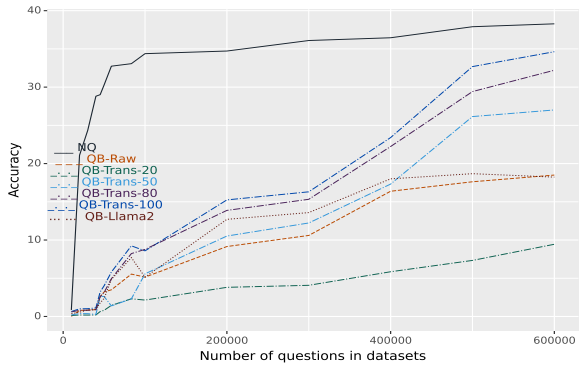

Figure 3: **REPLUG**: Again, **QB-trans** without any NQ data comes within 5 points of a model trained on **NQ**. **QB-Trans-50** comes within 5 points of a model trained on **QB-TRANS**.

.

using the syntactic rules, the transformed QB question becomes "in one of which man's paintings, one character oddly uses her left hand to grasp the red-cloaked character's chin while her right hand sits at his knee". This question (based on its length) would score poorly on the classifier, but nonetheless the answer is Jean Auguste Dominique Ingres. However, in the **QB-Llama2**, the question becomes "What is the significance of the left-hand grasp and the right-hand placement in Jean Auguste Dominique Ingres' painting featuring a red-cloaked character and another woman?" Not only does the desired answer change (it's not clear that there is a correct answer), but the answer appears in the question (despite the instructions in the prompt).

5 Training a Supervised QA System

We will be comparing all of our generated datasets with the original NQ dataset (**NQ**). While the NQ questions were selected based on some heuristics,[6] our QB questions are generated based on the heuristics described in Section 3; we combined the two datasets to construct QBANDNQ.

5.1 Supervised QA systems

As the baseline, we used state-of-the-art model in the NQ challenge leaderboard **ReflectionNet** (Wang et al., 2020) which consists of a MRC model for answer prediction and Reflection model for answer confidence. We also used **GENREAD** (Yu et al., 2023) which is a *generate-then-retrieve* pipeline QA system that directly generates the contextual documents by using clustering document representations. This method outperforms traditional *retrieve-then-read* pipeline methods. We also use the two retrieval based systems DPR(Karpukhin et al., 2020b) and REPLUG (Shi et al., 2023b) described in the previous section but this time trained with QB data along with NQ dataset.

5.2 Training Data

We train the supervised QA systems with our QBANDNQ dataset, combination of original NQ and QB questions. We also replace Yu et al. (2023) with QBANDNQ dataset to see how our dataset performs when merged with the NQ dataset and whether our dataset can be used as an expansion of the NQ dataset.

Like in the previous zero-shot experiment, we use different percentage of NQ questions along with QB generated questions for training the model. For example, **QB-NQ-10**, represents all of the filtered and transformed QBANDNQ data set and ten percent of the original NQ data.

We also transform answers from the QB dataset to look like the NQ data. For example, one of the QB question after transformation *Which ethnic group's language and customs were adopted by a majority of the uru people?* with answer *Aymara people (the Quechua were the larger group targeted by the genocide)*. However, if we observe the NQ answer list, there is no description given using the parenthesis. Therefore, we have converted the answer set to also include *Aymara people* to make the answer set look like NQ formatted. Uniformly at random selects transformed sentences without regard to how similar they are to natural questions. Sorted by the classifier is a deterministic order where all examples are processed in order of their classifier score (e.g., the best scoring transformation "Who coined the term "behaviorism"?" .is the first, the worst scoring—after 4000 examples—" Which country's capital sits on a namesake gulf jutting out from the south china sea ." is the last). In all cases, NQ examples are selected uniformly at random.

Finally, because a some data can go a long way, we also compare against combinations of NQ and our transformed sentences.

5.3 Result and Analysis

We had argued that using transformed QB data would be cheaper than using NQ data (which is expensive) to gather answers for. What if we have

[6] For example, the questions start with "who", "when" or "where" followed by a finite form of "do" or a modalverb (Kwiatkowski et al., 2019)

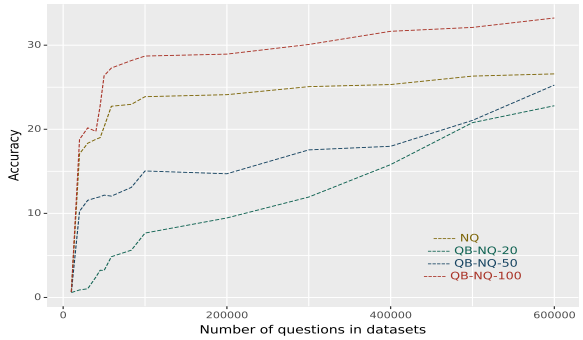

Figure 4: **DPR**: As expected, supervised training on QB-NQ-100 and evaluating on NQ produces the highest accuracy system with DPR. However, the cheaper datasets from our systematic conversion (**QB-NQ-50**), with a noiser but larger dataset, reaching a substantial fraction of the accuracy.

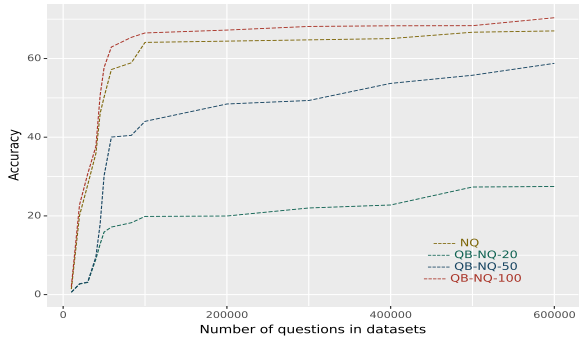

Figure 5: **ReflectionNet-Ensemble**: Again, in supervised setting, **QB-NQ-100** data crosses the **NQ** by 3 points of a model trained on **NQ**, and adding just 50% of the NQ data allows the model to reach within 5 points of the accuracy of the model trained on the whole **NQ** dataset.

access to a *fraction* of the NQ data? Finally, given the best configuration of the previous experiment, we add small amounts of NQ data to see how much is needed to recreate the best NQ result. Adding half of the NQ brings parity to the result. Therefore, our experiments show the effectiveness of QB question as an alternative of NQ dataset in the zero-shot setting and an expansion of NQ dataset in supervised QA systems. Similar results can be seen in all the systems. We have included DPR and ReflectNet-Ensemble here (Figure 4 and 5). ReflectionNet-Ensemble has higher accuracy than DPR because of its usage of ensemble model in training. No data in the training process is changed.

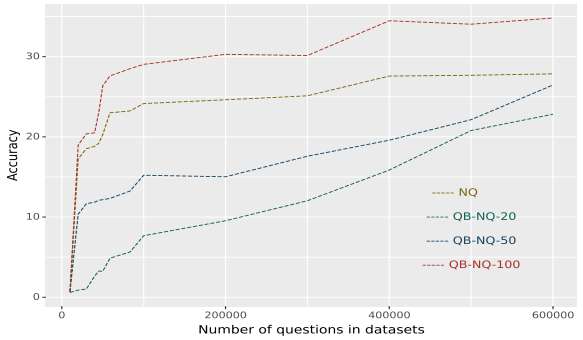

Figure 6: **DPR**: Again, **QB-NQ-100** data crosses by 5 points of a model trained on **NQ**, and adding just 50% of the NQ data allows the model to reach within 10 points of the whole NQ with answer equivalence.

6 Answer Equivalence in Zero-shot and Supervised Training

Thus far, we have focused on ensuring that the transformed questions resemble the target NQ data as much as possible, but have not considered the answers. To fully emulate NQ data, the answers need to be comparable. Thus, we expand the answer set provided in the QB dataset (which typically is more formal and verbose than NQ) with the WikiData answer equivalence sets from Si et al. (2021) for both training and evaluation. For example, NQ has a question "where do the greasers live in the outsiders?" with the correct answer set comprised of {'Tulsa , Oklahoma'}. However, if the QA system answers 'tulsa, oklahoma', it will be considered as incorrect in the exact match. Thus, we apply an answer equivalence system to change the answer set to {'Tulsa , Oklahoma', 'ttown', 'tulsa', 'tulsa oklahoma', 'wagoner county tulsa city'}.

After adding answer equivalence, the accuracy increased in both supervised and zero-shot setting (consistent with results in Si et al. (2021)), and while the gap in accuracy is still around five points, the percentage accuracy between **QB-trans** and **NQ** is much closer (Figure 6).

7 Analysis of Transformed Questions

Not all of the original elicitations are transformed correctly. First, there is transformation error. Take this original elicitation:

> The protagonist is rescued by Robert Walton in the Arctic and hails from Lake Geneva.

The first heuristic that is applied here is the split of clause based on conjunction. After applying the heuristic based on conjunction "and", we get

two clauses: "The protagonist is rescued by Robert Walton in the Arctic" and "The protagonist hails from Lake Geneva". Next, we add wh-words to produce questions: "Where the protagonist is rescued by Robert Walton in the Arctic?" and "Where the protagonist hails from Lake Geneva". These create poor questions that change the meaning of the original elicitation, but after applying the classifier, it nonetheless gets high score because it has features (e.g, similar length to NQ questions, begins with "where", and void of QB question patterns) similar to the NQ questions. These features lead to classify the question as close to NQ (Table 5).

8 Related Work

8.1 An Explosion of Datasets

The last few years has seen a flurry of datasets. Some of these datasets are created at great expense through crowdsourcing to capture common sense, numerical reasoning, visual QA (Antol et al., 2015), video QA (Yang et al., 2003), common sense questions (Talmor et al., 2021) or multicultural questions (Clark et al., 2020); Rogers et al. (2023) gives a thorough summary. Less common are datasets focusing on found data, although there are nonetheless a panoply of questions harvested from educational resources, civil service exams, users, and trivia games.

8.2 Generating Questions

Given the expense of gathering these data, an obvious alternative is to generate your data. While we transform one question format into another, Probably Asked Questions (Lewis et al., 2021)[PAQ] transforms source documents into questions that *could* be asked. These questions are more formulaic than the questions carefully crafted by trivia experts in the QB dataset, but an obvious extension would be to see if PAQ questions could help augment the results here. Another class of transformed questions are translated questions that convert datasets like SQUAD into multiple languages (Carrino et al., 2020; d'Hoffschmidt et al., 2020).

Given all of these datasets, a frequent research thrust has been to create methods to generalize from one QA setting to another, either by merging datasets together (Artetxe et al., 2019; Khashabi et al., 2020) or by QA-driven slot-filling (Du et al., 2021) or event extraction via QA (Lyu et al., 2021) by creating algorithms that explicitly generalize (Munteanu et al., 2004; Munteanu and Marcu, 2005).

8.3 Transforming Questions

Our approach of transforming the form of QB elicitations is inspired by a long line of research. Pre-neural QA work used machine translation models to transform questions into something that would resemble the text where the answer would be found (Wang et al., 2007). Other work transforms questions to remove ambiguity or to transform a context-dependent question into a question that more closely resembles NQ (Demszky et al., 2018).

8.4 Zero-shot QA

In zero-shot setting, large language model is used to generate new questions. In BeamSearchQA (Sun et al., 2023), new questions are generated using LLM by iterative refining and expanding scope of the question achieves a state of the art EM score 38.0, there are some approaches without the retriever. In-context learning approach is applied using GPT-3 (Brown et al., 2020), cost efficient Generalist Language Model (GLaM) GPT-3 (Du et al., 2022), instruction-tuned model (Wei et al., 2021) in zero-shot setting. Self-supervised knowledge learning is applied in zero-shot QA, for example heuristic based graph (Banerjee and Baral, 2020). However, in our work, we are creating nq-like questions from qb questions. The main difference of our work from the previous work is that, we are using a different dataset to train the model in a zero-shot to make it compatible with NQ dataset. With a proper classifier and carefully chosen heuristics, we introduce a conversion of different domain dataset as a replacement of NQ dataset.

9 Conclusion and Future Work

Transformed NQ-like questions from the QB data is an alternative to expensive datasets like NQ. The transformed data itself is not as good as NQ by itself, but is competitive; this is a reasonable option if the resources are not available to curate a dataset like NQ. If there is budget to create a dataset comparable to NQ, a small ammount of this data augmented with transformed data from a dataset like QB can surpass a model trained on the NQ dataset alone. For future work, we can apply this conversion technique for other languages where transformation heuristics can be learned using human data.

10 Limitations

Focus on Natural Questions We focus on NQ, a popular and respected dataset. Other datasets are different, and we do not know how well our transformations would generalize to other datasets. However, we suspect that similar transformations would also succeed.

Errors hidden by Correct Answers While our transformed data often gets to the right answer, we have not systematically verified that the produced questions are themselves correct. It could be that enough of the necessary contents within the conversions remain that systems can reach the correct answer but that the questions contain errors (either factual or grammatical). From our inspection of the questions, we do not believe this to be the case, but a systematic evaluation would be needed to confirm this. However, this would dramatically raise the cost of the dataset, obviating one of the motivations for this approach.

Distribution Shift QB and NQ have very different distributions: QB is more academic, while NQ has more questions about sports and pop culture. Thus, solely evaluating on NQ potentially says little about how well our conversion process works for the topics that are over-represented in QB compared to NQ. While NQ does have some questions about literature and science, they are under-represented; it could be that our transformations are particularly brittle on questions about equations or works of fiction but NQ evaluation does not expose that weakness.

Ethical Considerations

The most important ethical consideration of this paper is that we are using the data from the trivia community to train a model. In contrast to datasets like SearchQA (Dunn et al., 2017) or TriviaQA (Joshi et al., 2017) where it is unclear how the original trivia authors feel about the use of the data, the QB community explicitly welcomes the sharing and dissemination of the data to train QB players: datasets are covered by a creative commons license (and the norm of sharing indeed predates the formal creation of creative commons). While computer QA systems are a different kind of trivia player (machine rather than human), we believe that this would be in the spirit of the community.

References

Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. 2015. Vqa: Visual question answering. In *Proceedings of the IEEE international conference on computer vision*, pages 2425–2433.

Mikel Artetxe, Sebastian Ruder, and Dani Yogatama. 2019. On the cross-lingual transferability of monolingual representations. *CoRR*, abs/1910.11856.

Pratyay Banerjee and Chitta Baral. 2020. Self-supervised knowledge triplet learning for zero-shot question answering. *arXiv preprint arXiv:2005.00316*.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer.

Jordan Boyd-Graber and Benjamin Börschinger. 2020. What question answering can learn from trivia nerds. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7422–7435, Online. Association for Computational Linguistics.

Jordan Boyd-Graber and Benjamin Börschinger. 2020. What question answering can learn from trivia nerds. In *Association for Computational Linguistics*.

Jordan Boyd-Graber, Brianna Satinoff, He He, and Hal Daume III. 2012. Besting the quiz master: Crowdsourcing incremental classification games. In *Proceedings of Empirical Methods in Natural Language Processing*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Casimiro Pio Carrino, Marta R. Costa-jussà, and José A. R. Fonollosa. 2020. Automatic Spanish translation of SQuAD dataset for multi-lingual question answering. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 5515–5523, Marseille, France. European Language Resources Association.

Jonathan H Clark, Eunsol Choi, Michael Collins, Dan Garrette, Tom Kwiatkowski, Vitaly Nikolaev, and Jennimaria Palomaki. 2020. Tydi qa: A benchmark for information-seeking question answering in typologically diverse languages. *Transactions of the Association for Computational Linguistics*, 8:454–470.

David R Cox. 1958. The regression analysis of binary sequences. *Journal of the Royal Statistical Society: Series B (Methodological)*, 20(2):215–232.

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

Martin d'Hoffschmidt, Wacim Belblidia, Tom Brendlé, Quentin Heinrich, and Maxime Vidal. 2020. Fquad: French question answering dataset. *arXiv preprint arXiv:2002.06071*.

Nan Du, Yanping Huang, Andrew M Dai, Simon Tong, Dmitry Lepikhin, Yuanzhong Xu, Maxim Krikun, Yanqi Zhou, Adams Wei Yu, Orhan Firat, et al. 2022. Glam: Efficient scaling of language models with mixture-of-experts. In *International Conference on Machine Learning*, pages 5547–5569. PMLR.

Xinya Du, Luheng He, Qi Li, Dian Yu, Panupong Pasupat, and Yuan Zhang. 2021. Qa-driven zero-shot slot filling with weak supervision pretraining. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 654–664.

Matthew Dunn, Levent Sagun, Mike Higgins, V Ugur Guney, Volkan Cirik, and Kyunghyun Cho. 2017. Searchqa: A new q&a dataset augmented with context from a search engine. *arXiv preprint arXiv:1704.05179*.

C. Fellbaum. 1998. *WordNet : An Electronic Lexical Database*, chapter A semantic network of English verbs. MIT Press, Cambridge, MA.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc.

He He, Jordan Boyd-Graber, Kevin Kwok, and Hal Daumé III. 2016. Opponent modeling in deep reinforcement learning. In *International conference on machine learning*, pages 1804–1813. PMLR.

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

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020a. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.

Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen tau Yih. 2020b. Dense passage retrieval for open-domain question answering.

Daniel Khashabi, Sewon Min, Tushar Khot, Ashish Sabharwal, Oyvind Tafjord, Peter Clark, and Hannaneh Hajishirzi. 2020. Unifiedqa: Crossing format boundaries with a single qa system. *arXiv preprint arXiv:2005.00700*.

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.

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

Patrick Lewis, Yuxiang Wu, Linqing Liu, Pasquale Minervini, Heinrich Küttler, Aleksandra Piktus, Pontus Stenetorp, and Sebastian Riedel. 2021. Paq: 65 million probably-asked questions and what you can do with them. *Transactions of the Association for Computational Linguistics*, 9:1098–1115.

Qing Lyu, Hongming Zhang, Elior Sulem, and Dan Roth. 2021. Zero-shot event extraction via transfer learning: Challenges and insights. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 322–332.

Inbal Magar and Roy Schwartz. 2022. Data contamination: From memorization to exploitation. *arXiv preprint arXiv:2203.08242*.

Sewon Min, Julian Michael, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2020. AmbigQA: Answering ambiguous open-domain questions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5783–5797, Online. Association for Computational Linguistics.

Dragos Stefan Munteanu, Alexander Fraser, and Daniel Marcu. 2004. Improved machine translation performance via parallel sentence extraction from comparable corpora. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004*, pages 265–272.

Dragos Stefan Munteanu and Daniel Marcu. 2005. Improving machine translation performance by exploiting non-parallel corpora. *Computational Linguistics*, 31(4):477–504.

Arvind Narayanan. 2023. Gpt-4 and professional benchmarks: the wrong answer to the wrong question. `https://www.aisnakeoil.com/p/gpt-4-and-professional-benchmarks`.

Joakim Nivre. 2010. Dependency parsing. *Language and Linguistics Compass*, 4(3):138–152.

Iker García-Ferrero Julen Etxaniz Eneko Agirre Oscar Sainz, Jon Ander Campos. 2023. Did ChatGPT cheat on your test? https://hitz-zentroa.github.io/lm-contamination/blog/.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

Jiezhong Qiu, Hao Ma, Omer Levy, Wen-tau Yih, Sinong Wang, and Jie Tang. 2020. Blockwise self-attention for long document understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2555–2565, Online. Association for Computational Linguistics.

Pedro Rodriguez, Shi Feng, Mohit Iyyer, He He, and Jordan Boyd-Graber. 2021. Quizbowl: The case for incremental question answering.

Anna Rogers, Matt Gardner, and Isabelle Augenstein. 2023. Qa dataset explosion: A taxonomy of nlp resources for question answering and reading comprehension. *ACM Computing Surveys*, 55(10):1–45.

Weijia Shi, Anirudh Ajith, Mengzhou Xia, Yangsibo Huang, Daogao Liu, Terra Blevins, Danqi Chen, and Luke Zettlemoyer. 2023a. Detecting pretraining data from large language models.

Weijia Shi, Sewon Min, Michihiro Yasunaga, Minjoon Seo, Rich James, Mike Lewis, Luke Zettlemoyer, and Wen-tau Yih. 2023b. Replug: Retrieval-augmented black-box language models. *arXiv preprint arXiv:2301.12652*.

Chenglei Si, Chen Zhao, and Jordan Boyd-Graber. 2021. What's in a name? answer equivalence for open-domain question answering. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9623–9629, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Hao Sun, Xiao Liu, Yeyun Gong, Anlei Dong, Jingwen Lu, Yan Zhang, Daxin Jiang, Linjun Yang, Rangan Majumder, and Nan Duan. 2023. Beamsearchqa: Large language models are strong zero-shot qa solver.

Alon Talmor, Ori Yoran, Ronan Le Bras, Chandra Bhagavatula, Yoav Goldberg, Yejin Choi, and Jonathan Berant. 2021. CommonsenseQA 2.0: Exposing the limits of ai through gamification. In *Proceedings of Advances in Neural Information Processing Systems*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

A. M. Turing. 1950. I.—COMPUTING MACHINERY AND INTELLIGENCE. *Mind*, LIX(236):433–460.

Ellen M. Voorhees. 2019. pages 45–69. Springer International Publishing, Cham. [link].

Mengqiu Wang, Noah A Smith, and Teruko Mitamura. 2007. What is the jeopardy model? a quasi-synchronous grammar for qa. In *Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL)*, pages 22–32.

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.

Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. 2021. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*.

Hui Yang, Lekha Chaisorn, Yunlong Zhao, Shi-Yong Neo, and Tat-Seng Chua. 2003. Videoqa: question answering on news video. In *Proceedings of the eleventh ACM international conference on Multimedia*, pages 632–641.

Wenhao Yu, Dan Iter, Shuohang Wang, Yichong Xu, Mingxuan Ju, Soumya Sanyal, Chenguang Zhu, Michael Zeng, and Meng Jiang. 2023. Generate rather than retrieve: Large language models are strong context generators.

Xinyan Velocity Yu, Sewon Min, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2022. Crepe: Open-domain question answering with false presuppositions. *arXiv preprint arXiv:2211.17257*.

Manzil Zaheer, Guru Guruganesh, Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and Amr Ahmed. 2021. Big bird: Transformers for longer sequences.

Hang Zhang, Yeyun Gong, Yelong Shen, Weisheng Li, Jiancheng Lv, Nan Duan, and Weizhu Chen. 2021. Poolingformer: Long document modeling with pooling attention. In *International Conference on Machine Learning*, pages 12437–12446. PMLR.

A Heuristics List

Through observation of the linguistic and grammatical style of NQ we add additional heuristics to further improve the candidates such as **removing punctuation** and **adding subject**:

- **Removing punctation**: Natural questions typically do not include punctuation, so we remove punctuation at the boundary of a generated question.
- **Adding subject**: If a question is missing a subject (e.g., "wrote *Burmese Days*", we add "which" `answer_type` (in this example, author) to the beginning of the question.

A.1 What is a zero-shot system?

Zero-shot systems enables the models to answer the questions without explicitly trained on them. Under zero-shot setting for the NQ dataset, there can be no training on NQ data– not with questions and their answers and not with their contexual documents. Therefore, when given any NQ test data, the zero-shot systems directly encode the given question and predict the answer. A question q is given to the model as the input. Based on that input, the model generates the answer a denoted by $p(a|p, \theta)$ where θ is the model parameters (Yu et al., 2023).

The state-of-the-art zero-shot QA system ALLIES (Sun et al., 2023) framework generates additional questions through an iterative process. In this process an LLM is used to generate queries based on existing query-evidence pair and score the answer. This iteration process continues until the score reaches a predefined threshold. Therefore, this system decomposes the original question into multiple sub-questions and achieves state of the art performance on zero-shot setting for NQ dataset. Another state-of-the-art zero-shot model GENREAD Yu et al. (2023) uses large language model InstructGPT (Ouyang et al., 2022) to directly generate contextual documents from a given question.

A.2 DPR Training

The passages that contain any of the answer strings are positive examples, while the passages that do not are negative examples. One example is shown in Table 6.

.1 Large Language Models and Transformer-based Models

Due to the increasing sequence length, transformer uses sparse attention to handle the complexity of long document modeling (Zhang et al., 2021). In this method, each token is made to attend more important context or local context (Qiu et al., 2020). Another approach uses sliding window pattern to capture local information that includes Longformer (Beltagy et al., 2020), BigBird (Zaheer et al., 2021). Lastly, PoolingFormer (Zhang et al., 2021) uses full self-attention into two-level attention schema–first one works as a sliding window attention pattern and the second level increases receptive field. Wang et al. (2020) uses machine reading comprehension (MRC) model for answer prediction and a Reflection model for answer confidence. This achieves state-of-the-art performance on the NQ dataset in the leaderboard of NQ challenge.

Heuristic	Purpose	Example before Heuristic	Example after Heuristic
substitute non answer pronouns	Substitute non answer pronouns to noun+possesion.	she founded Carthage and reigned as its queen from 814-759 BC	she founded Carthage and reigned as carthage 's queen from 814-759 BC
clean marker	Remove punctuation patterns at the beginning and the end of the question.	which german philosopher is this philosopher wrote a work , . "	which german philosopher also wrote glowing reviews of which german philosopher's own works in ecce homo
drop after semicolon	Remove contents after semicolon in NQlike.	which molecule is this compound 's presence can be quantified in spectrophotometry by observing an intense absorption peak at 255 nanometers ; that peak is the	which molecule 's presence can be quantified in spectrophotometry by observing an intense absorption peak at 255 nanometers
convert continuous to present	Change the first verb to normal tense if it is in continuous tense.	which particle consisting of a charm quark and an anti - charm quark	which particle consists of a charm quark and an anti - charm quark
fix no wh words	Convert "this" to "which"+answer_type when there's no "wh-" words.	this play begins with the protagonist arriving at the elysian fields to see her sister stella	which play begins with the protagonist arriving at the elysian fields to see her sister stella
replace this is	Replace "this" to "which"+answer_type within "this is" pattern.	this is the first party name , followed by kraemer , in that supreme court case , which held that racially restrictive covenants are unconstitutional	which name the first party name , followed by kraemer , in that supreme court case , which held that racially restrictive covenants are unconstitutional
replace which with that	Convert "which" to "that" and check if no "which" present anymore, if so, convert "this" to "which".	michael green is a current professor at this university , which is where watson and crick discovered dna 's structure	michael green a current professor at which university , that is where watson and crick discovered dna 's structure

Table 2: List of Heuristics

Heuristic	Purpose	Example before Heuristic	Example after Heuristic
add question word	Adding "which"+answer_type when no "wh-" words present.	a chamberlain named cleander was killed on the orders of marcia , a mistress of this man who was involved in the plot that eventually assassinated him and replaced him with pertinax	a chamberlain named cleander killed on the orders of marcia , a mistress of which man who was involved in the plot that eventually assassinated him and replaced him with pertinax
add subject	Add "which"+answer_type at the beginning when question starting with VERB/AUX and missing the subject.	were refused real employment because of " logical discrimination , " an excuse which belied the employers ' fear of their " death taint	which se people were refused real employment because of " logical discrimination , " an excuse which belied the employers ' fear of their " death taint
fix what is which	Remove "what is" from "what is which".	what is which desert lying mostly in northern china and mongolia	which desert lying mostly in northern china and mongolia
remove end BE verbs	Remove "is/are" at the end of NQklike questions.	which jewish holiday is that hymn is	which jewish holiday is that hymn
remove extra AUX	Remove extra auxiliary words.	which number is it is the base for solutions to the differential equation	which number is the base for solutions to the differential equation
remove patterns	Remove bad patterns in NQlike.	which irish playwright is andrew (*) undershaft	which irish playwright is andrew undershaft
remove rep subject	remove repetition of the subject “is this”.	which goddess is this goddess is considered a daughter of ra	which goddess is considered a daughter of ra
remove BE determiner	Change is his/is her/is its to ’s.	which greek goddess’s is her wedding night lasted three hundred years	which greek goddess’s wedding night lasted three hundred years
remove repeated pronoun	Removes repeated pronouns like "which character who is", "is who is".	which character who is the character who never appears to linus in a peanuts halloween special	which character never appears to linus in a peanuts halloween special

Table 3: List of Heuristics.

Heuristic	Purpose	Example before Heuristic	Example after Heuristic
fix no verb	Ensure there's at least one verb per question.	which greek god wielding chief greek god	which greek god is wielding chief greek god
add space before punctuation	Add space before punctuation because in NQ there's space before all types of punctuation	which greek goddess's wedding night lasted three hundred years	which greek goddess 's wedding night lasted three hundred years
rejoin whose	replace "who's" with "whose"	which wife who 's kidnapping by paris began the trojan war	which wife whose kidnapping by paris began the trojan war

Table 4: List of Heuristics.

Feature	Weight
percentile length >5	−5.49
bigram START how	−4.98
bigram did the	−3.99
bigram does the	−3.58
bigram of this	3.52
bigram which man	3.38
bigram how many	−3.38
bigram START this	3.20
bigram was the	−3.00
bigram of what	2.88
bigram in this	2.73
bigram when did	−2.42
bigram START when	−2.40
no QB pattern	−2.27
bigram START where	−2.24
bigram who plays	−2.19
bigram who played	−2.14
bigram of which	1.95
bigram START one	1.74

Table 5: To identify which generated questions most resemble our target information-seeking paradigm NQ questions (negative features) vs. the source probing domain, we run a simple classifier over bigrams and question statistics. The classifier prefers shorter generated questions, questions that begin with question words, and questions without QB idiosyncratic patterns (no QB pattern): stock phrases like "for 10 points", "name this", etc. The classifier is used to prioritize the data used to train later QA models.

Question	A fortification overlooking which city was renamed “narin qala” or “little fortress” by mongolinvaders in the 13th century.
Answer	Tbilisi
Positive context	City in the Caucasus, with its at least 50,000 inhabitants and thriving commerce. Several intellectuals born or living in Tbilisi, bearing the nisba al-Tiflisi were known across the Muslim world. The Abbasid Caliphate weakened after the Abbasid civil war in the 810s, and caliphal power was challenged by secessionist tendencies among peripheral rulers, including those of **Tbilisi**. At the same time, the emirate became a target of the resurgent Georgian Bagrationi dynasty who were expanding their territory from Tao-Klarjeti across Georgian lands. The Emirate of Tbilisi grew in relative strength under Ishaq ibn Isma’il, who was powerful enough to
Negative context	near the shores of Kasagh River, during the reign of king Orontes I Sakavakyats of Armenia (570̆2013560 BC). However, in his first book “Wars of Justinian”, the Byzantine historian Procopius has cited to the city as “Valashabad” (Balashabad), named after king “Valash” (Balash) of Armenia. The name evolved into its later form by the shift in the medial “L” into a “Gh”, which is common in the Armenian language. Movses Khorenatsi mentioned that the Town of Vardges was entirely rebuilt and fenced by king Vagharsh I to become known as “Noarakaghak” (,“New City”) and later “Vagharshapat”. The territory of

Table 6: We have a QB question: *A fortification overlooking which city was renamed “narin qala” or “little fortress” by mongolinvaders in the 13th century.* with answer *Tbilisi*. Now, for the positive context of the DPR training we have used those passage which contain the answer string and the rest of the passages are selected as negative context. One of the examples of positive contexts and negative contexts for this question is shown here.