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

Saptarashmi Bandyopadhyay
University of Maryland
saptab1@umd.edu

Hao Zou
University of Minnesota
zou00080@umn.edu

Chenqi Zhu
University of Maryland
chqzhu@terpmail.umd.com

Abhranil Chandra
IIT Kharagpur
abhranil.iitkgp@gmail.com

Jordan Boyd-Graber
University of Maryland
jbg@umiacs.umd.edu

Abstract

Training question answering (QA) systems for web queries require large, expensive datasets that are difficult to annotate and time-consuming to gather. However, for many languages, there are large enthusiast-generated datasets of questions and answers. Thus an opportunity presents itself: we automatically generate shorter, information-seeking questions, resembling web queries in the style of Natural Questions (NQ) dataset, from longer trivia questions. 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 features, to find transformed questions that match NQ in style and topic. We show that training a QA system on these transformed questions is a viable strategy for low resource settings.

1 Introduction

Question answering is a central problem in AI research. One way of understanding *why* people ask question was proposed by Rodriguez and Boyd-Graber (2021): questions come from either an information seeking paradigm (Voorhees, 2019, Cranfield) or an evaluation paradigm (TURING, 1950, Manchester). While it is very easy to get *questions* in the Cranfield 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 Manchester questions are fundamentally better because they lack ambiguity (Min et al., 2020) and are more artfully crafted.

However, these bold claims have not been supported by hard evidence. Other than exhibition matches (Ferrucci et al., 2010), the community has gravitated to Cranfield-paradigm questions from NIST, Microsoft, and Google. These datasets are more expensive than their Manchester counterparts, which are for the most part written for free by trivia enthusiasts in many languages.

This paper investigates whether we can transform the idiosyncratic, unrealistic questions from one trivia community (Section 2) into questions that look like more natural questions. Such a process, rather than requiring expensive annotations can be done with rule-based transformations (Section 3) without complicated machine learning. We then select the most natural questions from those transformed questions using a quality classifier (Section 4) to create a system to evaluate on real Natural Questions (Section 5).

2 An Artful but Arcane Trivia Dataset

Consider a typical question in the 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?

this question is much longer than the kind of question you would typically see asked of Siri or Alexa. That is because it is designed to be interruped 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.

Thus, approaches to answer QB questions directly are complicated because they must decode the convoluted syntax ("moved his capital from Kraków to this city on the Vistula"), decide not just what to answer, but also *when* to answer (**?**).

Our goal is to avoid this baroque complexity and use what is actually useful in the QB format: a series of pieces of 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. Each of these could become a standalone question.

Thus, our goal is to turn each of these facts into something that looks like a Natural Question (Kwiatkowski et al., 2019), a dataset collected by Google from real questions people have asked online. These questions are substantially shorter, typically only a handful of words, and have an answer annotated from a Wikipedia page.

At first blush, the released QB and NQ datasets seem comparable: QB has a total of 119247 question/answer samples and NQ has 91434. But these comparable topline numbers belie the substantial differences underneath in cost, quality, and quantity.

First, while the QB questions are unambiguously paired with the answer by the author of the question, NQ questions must be laboriously annotated by paid workers. While Google has not officially released numbers, 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.

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 question 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 question would produce several clues about an entity, allowing a model to understand more about each potential answer.

3 Transforming into a Natural Question

Having motivated why we want to convert QB questions to NQ questions, this section outlines our method of converting the long QB questions into multiple relevant NQ-like questions. The example of the whole transformation of a QB question is illustrated in Figure 2.

3.1 Generating Candidates

Cannonical Answer Type The first step is to find a cannonical answer type for an answer. This is important because sometimes questions written in QB's pyradmidal style use oblique references: "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". We look over all QB questions and for each disambiguated answer find the most frequent string used to ask about the answer. These cannonical answer types then will replace the mentions in the original question.

Mention Detection and Candidate Extraction To find all the references to the answer, we run a coreference system (Kirstain et al., 2021) and find the cluster referring to the answer. Each of the references to the answer then becomes a possible stand-alone question. We take the minimal verb phrase that contains the mention as a the candidates to form candidate questions.

Conjunction and Relative Clause Normalization Given these candidates, we then need to extract the minimal facts that would form the basis of a question. For example, if the QB question 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 and a single verb.

Imperative to Interrogative The most obvious difference between QB and NQ questions is that QB questions aren't grammatical questions: rather, they are declaritive statements about the answer. 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".

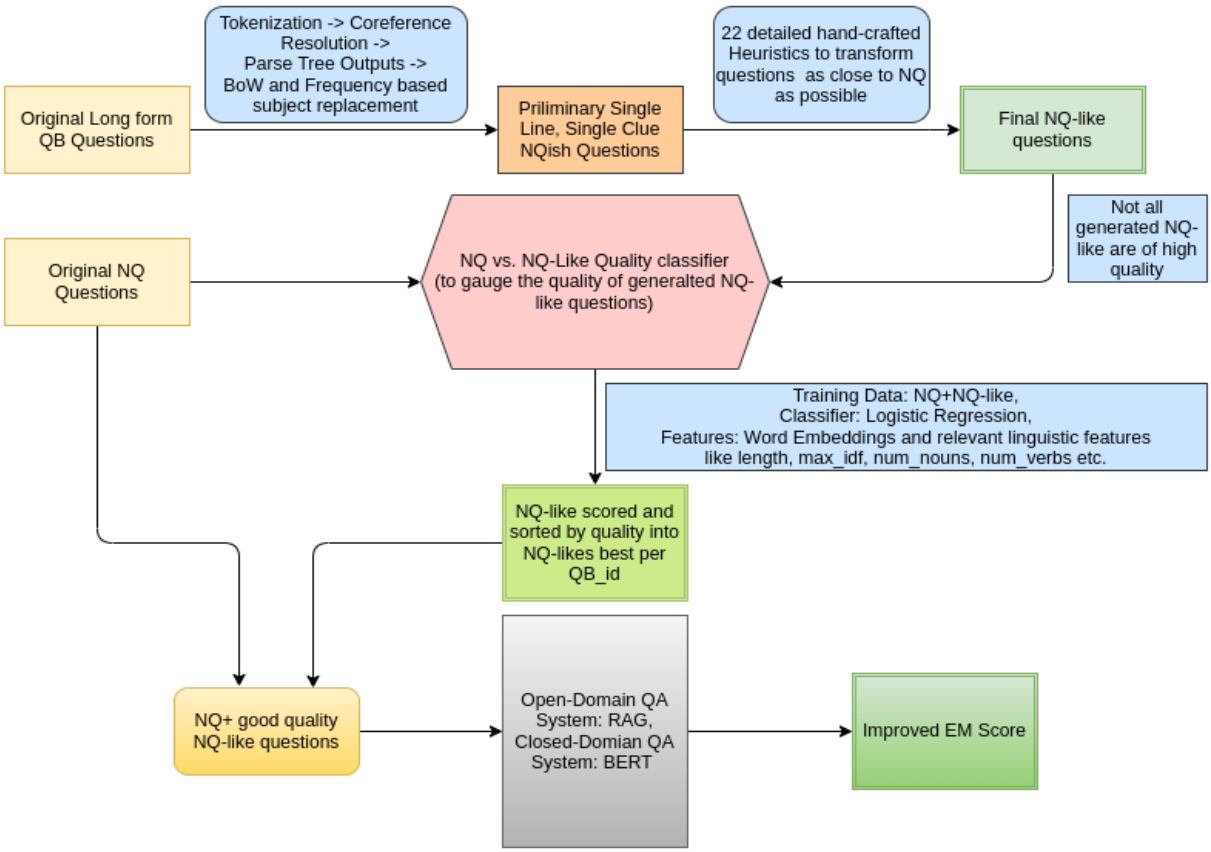

Figure 1: QB2NQ Pipeline

Identify Answer Type For an QA system to correctly find an answer, it is crucial to identify the answer type of the question(Lally et al., 2012). Therefore, the final step is to replace mentions with interrogative pronouns or "what" paired with the corresponding lexical answer type. In order to extract the lexical answer type from QB, we develop Algorithm 1 to recognize the most commonly referred word that is in the span of the nominal mention of the answer. Then we assign each QB *question* with a lexical answer type, which is later used in generating NQ-like. Continuing from earlier example, "who wrote Animal Farm" turns into "what author wrote Animal Farm", which is easier for QA system to identify the answer.

Additional Heuristics Through careful observation of the linguistic and grammatical style of NQ we add additional heuristics to further improve the candidates:

- **Substituting non-answer pronouns**: Substituting non-answer pronouns to noun + possesion.
- **Cleaning marker**: Remove hanging punctuation patterns at the beginning and the end of the questions.
- **Cleaning answer type**: Convert "– name this" patterns to "which".
- **Fixing no 'wh' words** : Convert "this" to "which"+answer_type when there's no "wh-" words.
- **Replacing this is** : Replace "this" to "which"+answer_type within "this is" pattern.
- **Adding question word** : Adding "which"+answer_type when no "wh-" words present.
- **Adding subject**: Add "which"+answer_type at the beginning when question starting with VERB/AUX and missing the subject.
- **Adding space before punctuation**: Add space before punctuation because in NQ there's space before all types of punctuation.

The detailed list of all the heuristics and their corresponding effect on the generated questions have been shown with examples in the Appendix B. We recursively apply all the heuristics to each of the primitive NQ-like question to generate as much NQ-like as possible and select the best ones based on Quality Control in the following section.

checked

Algorithm 1 Compute LAT Frequency

procedure COUNTANSWERTYPES($questoin, answer$)
  LAT $\leftarrow Counter()$ ▷ Initialize Counter.
  for $span \in NominalMentions(question, answer)$ **do**
    $mention \leftarrow span[1:]$
    LAT$[mention] \leftarrow$ **LAT**$[mention]+1$
  ▷ For each span of the nominal mention of the answer, count the number of mentions.
  end for
  return $MostCommon$(**LAT**)
end procedure
procedure COMPUTELAT(QB)
  LATS $\leftarrow$
  for $question, answer, ID \in QB$ **do**
    LATS$[ID] \leftarrow CountAnswerTypes(question, answer)$ ▷ Compute lexical answer type for each QB *question*
  end for
  return LATS
end procedure

Feature	Weight
percentile_length_5	−5.492091
bigram *START how*	−4.986225
bigram *did the*	−3.990464
bigram *does the*	−3.586583
bigram *of this*	3.525547
bigram *which man*	3.388073
bigram *how many*	−3.382850
bigram *START this*	3.209192
bigram *was the*	−3.003000
bigram *of what*	2.884738
bigram *in this*	2.738122
bigram *when did*	−2.421508
bigram *START when*	−2.406287
no_QB_pattern	−2.279292
bigram *START where*	−2.245444
bigram *who plays*	−2.193579
bigram *who played*	−2.143940
bigram *of which*	1.958976
bigram *START one*	1.743725

Table 1: Top 20 features by absolute weight.

3.2 Selecting Candidates

The above process generates many candidates, not all of which are good: some are too short, some are too long, some do not make sense, others still look too much like a QB question. Thus, we use a simple logistic regression classifier(Brzezinski, 2000) with a small set of features to select the candidates that are most like NQ examples: we train the classifier to distinguish our candidate from NQ questions.

We conduct feature engineering(Kuhn and Johnson, 2019) and specifically use a simple feature set so that we do not overfit and to not require a large training set. The entirety of features we have experimented and their definitions are listed in Appendix A, with the features we select highlighted. We only merge an equal number of 104071 examples from both QB and NQ to form the training data. The training and evaluation process is detailed in Algorithm 3. Then the candidates that most resemble NQ questions are selected (at most one per original sentence in a QB question).

While we are using NQ *questions*, we are critically not using the answers to the questions, which (unlike the answers) are expensive to collect for NQ. Nonetheless, some of our features help identify topics of questions that occur frequently in NQ by looking at the feature weights as listed in Table 1.

For example:

- "who played" is highly weight features, which account for NQ's high pop-culture proportion.
- "which man" is also a highly weighted feature as many NQ questions are asking for historical figures/celebrities.
- Starting with a question word like "how", "when", "where" are bestowed high weights as most of NQ starts with them while starting with word like "this" or "one" have high weights too but in positive value as QB often start with them.
- percentile_length_5 has the largest weight, which suggests being short is critical to for a question to be NQand no_qb_pattern feature having a high weight is self-evident since naturally NQ doesn't have patterns prevalent in QB.

3.3 Evidence selection strategy

Step 1: We have 104071 NQ questions with 63833 unique answers. We have 2.1 million NQ-like ques-

tions with 36999 unique answers.

Step 2: For every answer, we select the context from the wiki page

Step 3: Check if there are multiple paragraphs containing the answer

Step 4: If there are multiple paragraphs containing the answer, take a cosine similarity (makes sense) of the question and paragraph, select the specific paragraph with highest similarity score as context (EVIDENCE SELECTION STRATEGY)

Step 5: If there is 1 paragraph only containing the answer, use it as the context.

Algorithm 2 Evidence selection

procedure EVIDENCESELECTION($questoin, answer$)
  LAT $\leftarrow Counter()$ ▷ Initialize Counter.
  for $span \in NominalMentions(question, answer)$ **do**
    $mention \leftarrow span[1:]$
    LAT$[mention] \leftarrow$ **LAT**$[mention]+1$
  ▷ For each span of the nominal mention of the answer, count the number of mentions.
  end for
  return $MostCommon$(**LAT**)
end procedure
procedure COMPUTELAT(QB)
  LATS $\leftarrow$
  for $question, answer, ID \in QB$ **do**
    LATS$[ID] \leftarrow CountAnswerTypes(question, answer)$ ▷ Compute lexical answer type for each QB *question*
  end for
  return LATS
end procedure

4 Experiments

4.1 Generated Data

We originally possess 105871 total samples for the NQ dataset and 112926 total samples for the QB full dataset, which consists of QB questions having multiple sentences. We generate 57739540 samples of NQ-like questions from QB questions through the procedures outlined in Section 3.1. We then extract 2169504 total samples of only the well-formed questions by utilizing a quality control classifier as outlined in Section 3.2. For each QB question, we also select the best well-formed NQ-like according to the same classifier, building a total sample of 112926 best questions per each QB ID. The NQ dataset is concatenated with the 4 different types of datasets outlined above and in Table 2. Assuming a generic *[Dataset Name]* for the 4 datasets, we name the augmented system like (NQ + *[Dataset Name]*) shown in Tables 3 and 4.

Algorithm 3 NQ vs. NQ-like Quality Classifier

procedure QUALITYCLASSIFIER($\underline{training\ data}, nq\_like$)
  $\mathbf{X} \leftarrow PrepareFeature(\underline{training\ data})$ ▷ Generate features for *training data*(consisting of ***NQ+NQ-like*** questions).
  $\mathbf{Y} \leftarrow GetLabels(\underline{training\ data})$ ▷ Obtain Labels for training data.
  Classifier $\leftarrow LogisticRegressionFitting(\mathbf{X}, \mathbf{Y})$ ▷ Train a Logistic Regression Classifier.
  Scores $\leftarrow ApplyClassifier($**Classifier**$, nq\_like)$ ▷ Evaluate NQ-like data on Classifier.
  return Scores
end procedure

4.2 Question Answering Systems

After reviewing several QA systems (Zhu et al., 2021), we finally used the Retrieval Augmented Generation (RAG) (Lewis et al., 2020b) for open-domain question answering and the Bert-based Extractive QA System (Devlin et al., 2019) for closed-domain question answering system as our baseline QA systems. We used a small subset of the Wikipedia dump as our retrieval dump for the low resource setting.

RAG QA System RAG is a hybrid, end-to-end differentiable model that combines an information retrieval component(non-parametric memory), ie. Wikipedia dense vector index from a pretrained neural retriever, the DRP(Karpukhin et al., 2020) part of RAG, with a seq2seq generator, ie. the BART(Lewis et al., 2020a) component of RAG(pretrained parametric memory). Instead of passing the input directly to the generator like a standard seq2seq model, RAG instead uses the input to retrieve a set of relevant documents. This approach enables researchers to efficiently control what RAG knows and doesn't know without wasting time on whole-model retraining

System Description	Data Size	Mean	Median	Mode
NQ system with full questions	104071	9.2	9.0	8.0
QB system with full questions	112926	112.1	110.0	110.0
Generated NQ-like full questions	57739540	14.0	16.1	3.0
Filtered generated NQ-like full questions	2169504	14.3	13.0	8.0
Best NQ-like per QB ID full questions	112926	10.3	9.0	6.0

Table 2: Statistics of number of words per Quizbowl question

Question Transformation

Original Full QB question

Chris Carney represents this state's 10th district in congress, which includes Snyder and Wyoming counties.
It is home to the nation's first zoo, and houses the Harry Houdini museum. It has the eastern hemlock as its state tree, the ruffed grouse as state bird, and Bloomsburg is the only officially incorporated town in this state. Its highest point is at mount Davies, and it includes Raystown Lake; the Monongahela ends in this state, where it meets the Allegheny river. Allentown and Reading are two of the larger cities in this commonwealth, and Bethlehem gave its name to a large steel company here. Scranton is the center of the coal mining industry in this rustbelt state, which is the starting point of the Ohio river. Also known as the Keystone state, and with capital at Harrisburg, FTP what northeastern state has Philadelphia as its metropolis, and is named after its Quaker founder?

Tokenization

1. Chris Carney represents this state's 10th district in congress, which includes Snyder and Wyoming counties.
2. It is home to the nation's first zoo, and houses the Harry Houdini museum.
3. It has the eastern hemlock as its state tree, the ruffed grouse as state bird, and Bloomsburg is the only officially incorporated town in this state.
4. Its highest point is at mount Davies, and it includes Raystown Lake; the Monongahela ends in this state, where it meets the Allegheny river.
5. Allentown and Reading are two of the larger cities in this commonwealth, and Bethlehem gave its name to a large steel company here.
6. Scranton is the center of the coal mining industry in this rustbelt state, which is the starting point of the Ohio river.
7. Also known as the Keystone state, and with capital at Harrisburg, FTP what northeastern state has Philadelphia as its metropolis, and is named after its Quaker founder?

Coreference Output

In Tokenization point 4: [Bethlehem, its]
In Tokenization point 6: [what northeastern state, its, its]

Parse-tree Output

1. Chris Carney represents this state 's 10th district in congress , which includes Snyder and Wyoming counties
2. It is home to the nation 's first zoo , andhouses the Harry Houdini museum
3. It has the eastern hemlock as its state tree , the ruffed grouse as state bird ,
4. Bloomsburg is the only officially incorporated town in this state
5. Its highest point is at mount Davies ,
6. it includes Raystown Lake ; the Monongahela ends in this state , where it meets the Allegheny river
7. Allentown and Reading are two of the larger cities in this commonwealth ,
8. Bethlehem gave Bethlehem's name to a large steel company here
9. Scranton is the center of the coal mining industry in this rustbelt state , which is the starting point of the Ohio river
10. Also known as the Keystone state ,
11. with capital at Harrisburg , FTP what northeastern state has Philadelphia as its metropolis , and is named after its Quaker founder ?

Bag of Words on last sentence with string replacement

with capital at Harrisburg , what northeastern state has Philadelphia as its metropolis , and is named after its Quaker founder ?

Bag of Words on the remaining sentences with word frequency based dictionary

Chris Carney represents which state 's 10th district in congress , which includes Snyder and Wyoming counties

What is home to the nation 's first zoo , and .houses the Harry Houdini museum

What has the eastern hemlock as what's state tree , the ruffed grouse as state bird ,

Bloomsburg is the only officially incorporated town in which state

It's highest point is at mount Davies ,

What includes Raystown Lake ; the Monongahela ends in which state , where what meets the Allegheny river

Allentown and Reading are two of the larger cities in which commonwealth ,

Bethlehem gave It's name to a large steel company here

Scranton is the center of the coal mining industry in which rustbelt state , which is the starting point of the Ohio river

Also known as the Keystone state ,

Top five questions after heuristics based on quality score.

Score	NQ-like question after heuristics
0.5892	which state is also known as the keystone state
0.5897	chris carney represents what state 's 10th district in congress
0.5963	which state's highest point is at mount davies
0.6196	allentown and reading are two of the larger cities in which commonwealth
0.9996	with capital at harrisburg what northeastern state has philadelphia as its metropolis , and is named after its quaker founder

Figure 2: Generation of NQ-like Questions from the original QB Questions.

System	Total samples	Accuracy	Precision	Recall	F1	SacreBLEU
NQ	1874	0.490	0.550	0.540	0.540	29.57
NQ + NQ-like from QB full	9140	0.390	0.420	0.410	0.420	27.95
NQ + Quality controlled NQ-like from QB full	3319	0.467	0.505	0.503	0.502	34.03
NQ + QB full	3092	0.465	0.503	0.502	0.499	27.62
NQ + QB last sentence	3092	0.516	0.558	0.560	0.556	37.37
NQ + NQ-like from QB last sentence	3259	0.500	0.557	0.556	0.553	34.98
NQ + Quality controlled NQ-like from QB last sentence	2967	**0.518**	**0.567**	**0.564**	**0.562**	**37.46**
QB full	1874	0.232	0.261	0.259	0.259	12.17
NQ-like from QB full	11062	0.171	0.188	0.188	0.188	12.45
Quality controlled NQ-like from QB full	2075	0.176	0.192	0.191	0.191	14.11
QB last sentence	1874	0.44	**0.495**	**0.488**	**0.488**	22.18
NQ-like from QB last sentence	2112	0.421	0.454	0.456	0.453	34.49
Quality controlled NQ-like from QB last sentence	1632	**0.441**	**0.495**	**0.488**	**0.488**	**38.25**

Table 3: Question Answering Metric values with RAG system

System	Total samples	Accuracy	Precision	Recall	F1	SacreBLEU
NQ	1874	0.211	0.243	0.242	0.241	8.74
NQ + NQ-like from QB full	9140	0.243	0.271	0.270	0.269	10.49
NQ + Quality controlled NQ-like from QB full	3319	0.227	0.256	0.254	0.254	**11.01**
NQ + QB full	3092	0.229	0.257	0.258	0.256	10.97
NQ + QB last sentence	3092	**0.257**	**0.291**	**0.295**	**0.290**	9.65
NQ + NQ-like from QB last sentence	3259	0.253	0.281	0.276	0.277	10.20
NQ + Quality controlled NQ-like from QB last sentence	2967	0.234	0.268	0.267	0.266	8.91
QB full	1874	0.191	0.251	0.228	0.233	7.17
NQ-like from QB full	11062	0.027	0.085	0.085	0.085	2.23
Quality controlled NQ-like from QB full	2075	0.197	0.224	0.218	0.219	8.71
QB last sentence	1874	0.328	0.394	0.384	0.383	9.66
NQ-like from QB last sentence	2112	0.310	0.363	0.358	0.357	**25.40**
Quality controlled NQ-like from QB last sentence	1632	**0.367**	**0.428**	**0.429**	**0.424**	23.16

Table 4: Question Answering Metric values with DrQA system

processes. RAG thus has two sources of knowledge: that which the seq2seq model(BART) store in their parameters and the knowledge stored in the corpus(dense vector index from retriever DPR). This setup is designed to combine the flexibility of "closed-book" (parametric-only) approaches with the performance of "open-book" or retrieval-based (non-parametric) approaches to enable RAG to excel at knowledge-intensive Natural Language Generation tasks.

We retrieve 5 documents using the RAG retriever and use a batch size of 2 for our training. We train our systems for 5 epochs by using AdaGrad (Duchi et al., 2011) as the optimizer with a learning rate of $1e^{-4}$.

Bert-based Extractive QA System In Extractive QA, the model extracts the answer from a context which could be a provided text, a table or even HTML. This is usually solved with BERT-like models. We trained teh QA system using teh bert-base-cased model from scratch on the questions, answers and context tuples that we have for QB dataset, NQ dataset and the transformed NQ-like generated questions dataset that we created. In the training procedure we follow the simple masked language modelling strategy and the made optputs are the start and end token of the possible answer form the context.

5 Results

Our research work demonstrates a clear improvement in QA performance for BERT and RAG. We see an improvement in QA evaluation scores when trivia questions with multiple sentences in QB dataset can be used to automatically generate NQ-like questions which are then concatenated with the original NQ questions. For the RAG QA system, the baseline score was improved by 3 of our proposed systems, out of which generated outputs from the last sentence concatenated with original NQ data has seen the most success with an improvement of 2.8 points in accuracy and an increase of 1.7, 2.4, 2.2 and 7.89 points on Precision, Recall, F1 measure, and BLEU respectively. The performance of the QA system trained on NQ-like filtered data concatenated with the NQ data is also better than the system based on the concatenation of QB

dataset with NQ dataset. For DrQA, all of our proposed systems have done better than the baseline with NQ data and the system with concatenated NQ and QB data. The generated outputs from the last sentence of the QB concatenated with NQ have shown an improvement of 4.6 points in accuracy and 4.8, 5.3, 4.9, 0.91 points in Precision, Recall, F1 measure, and BLEU respectively on the DrQA system. Most of our systems exhibit an increase in BLEU score over the baseline. We also observe that the performance of RAG and DrQA systems trained on only NQ like data generated from QB dataset is better than the baseline systems on QB dataset.

This is understandable as the last sentence usually has the easiest clue. On top of which, it has a regular structure. These characteristics make the last sentence from the QB more semantically aligned with NQ-like data after conversion using our algorithm.

Our proposed NQ-like generation as outlined in Section 3.1 and quality filtering technique (1093 extra samples) as mentioned in Section 3.2 can be effective in a low resource setting with a lack of sufficiently well annotated QA data. This also helps in scaling the training data by converting multiple datasets automatically.

A broad example of NQ-like question generation and subsequent answer generation has been provided in Figure 2.

6 Limitations

- Profiting on the work of a community
- Don't represent a NQ question distribution (dates and numbers are a big issue, less popular culture)
- Errors in transformation process

7 Conclusion and Future Work

We clearly observe from the results that adding filtered NQ-like questions from the QB data has given a boost over using only NQ questions. In finer detail, we observe that questions from the last sentence of the QB are of higher quality than from intermediate sentences and therefore provide a higher boost to performance even with less samples. Even by simply adding questions generated from last sentence, we increase the exact match accuracy by nearly 2 points. We also observe that the BLEU score of answers generated from quality controlled NQ like system is 16 points more than the BLEU score of the baseline QB system for the RAG system and by 13 points for the DrQA system. This shows that our algorithm to generate NQ-like questions has been effective in improving the quality of the training dataset.

We are working to extend this system from the smaller datasets to the entire 119247 QB and 91494 NQ samples by using the generated 772456 unfiltered NQ-like questions from the entire QB data, retain the well-formed questions and improve the performance over the baseline system trained on NQ dataset.

We observe that QB trivia questions are in passive voice while NQ questions are in active voice. So Neural Machine Translation can help in converting the style of QB questions. Our dataset pairing method automatically filters out adversarial samples and improves the quality of the generated NQ-like corpus. This opens the door towards a possible research area where multiple non-adversarial datasets can be used to filter the NQ dataset from adversarial samples. We plan on improving the filtering process by generating our own annotated dataset regarding well-formed and ill-formed questions in order to detect well formed questions and merge ill formed questions with better quality questions.

Specifically, through our manual transformation for NQ-like questions, we are specifying three different answer types like WHO/WHICH/WHAT, which might be useful to propagate to intermediate parse outputs during NQ-like question generation in order to improve answer generation.

Other information like difficulty and year associated with QB questions can be leveraged to improve quality or filtering and can even be used to predict extra information on NQ questions.

Further work can also be done on augmenting other datasets to NQ-like questions to generate more synthetic samples which may improve question answering performance.

References

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.

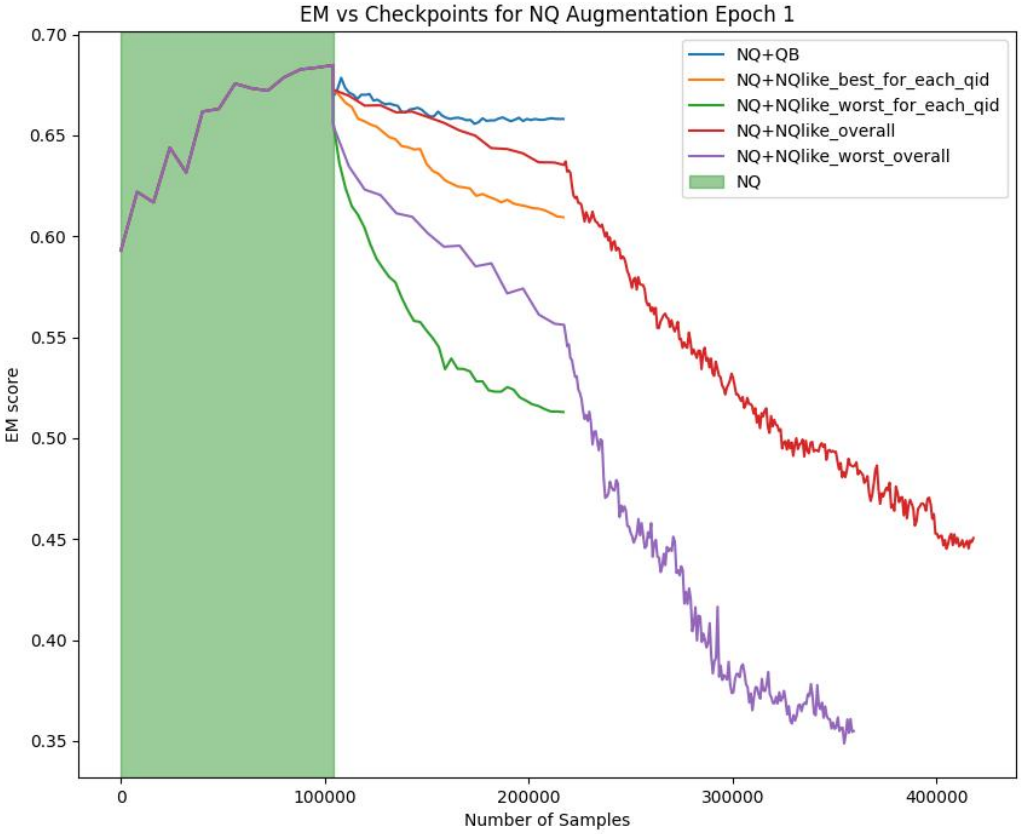

Figure 3: EM NQ Augmentation

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*.

Jacek R. Brzezinski. 2000. *Logistic regression for classification of text documents*. Ph.D. thesis, DePaul University, School of Computer Science, Telecommunications, and Information Systems.

K. Church and W. Gale. 1999. *Inverse Document Frequency (IDF): A Measure of Deviations from Poisson*, pages 283–295. Springer Netherlands.

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

John Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(Jul):2121–2159.

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

David Ferrucci, Eric Brown, Jennifer Chu-Carroll, James Fan, David Gondek, Aditya A. Kalyanpur, Adam Lally, J. William Murdock, Eric Nyberg, John Prager, Nico Schlaefer, and Chris Welty. 2010. Building Watson: An Overview of the DeepQA Project. *AI Magazine*, 31(3).

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. 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.

Najoung Kim, Ellie Pavlick, Burcu Karagol Ayan, and Deepak Ramachandran. 2021. Which linguist invented the lightbulb? presupposition verification for

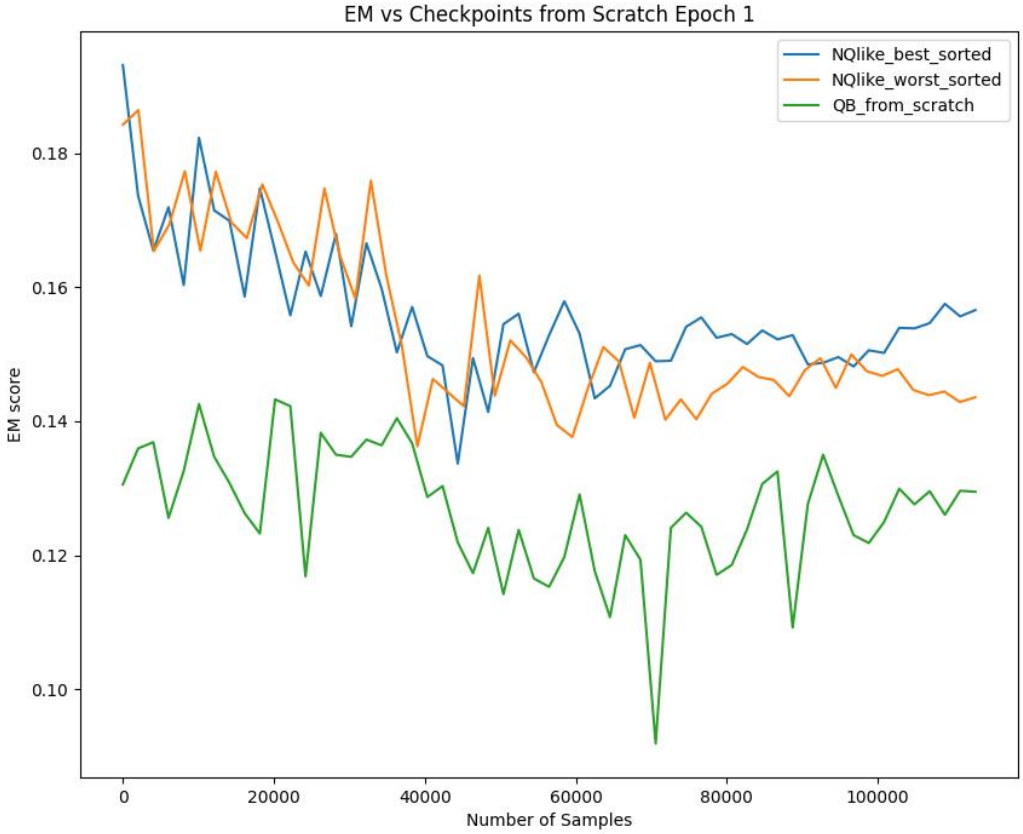

Figure 4: EM from Scratch

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.

J. Peter Kincaid, Robert P. Fishburne, R L Rogers, and Brad S. Chissom. 1975. Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel.

Yuval Kirstain, Ori Ram, and Omer Levy. 2021. Coreference resolution without span representations.

Grzegorz Kondrak. 2005. N-gram similarity and distance. In *String Processing and Information Retrieval*, pages 115–126.

Max Kuhn and Kjell Johnson. 2019. *Feature engineering and selection: A practical approach for predictive models*. CRC Press.

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.

A. Lally, J. M. Prager, M. C. McCord, B. K. Boguraev, S. Patwardhan, J. Fan, P. Fodor, and J. Chu-Carroll. 2012. Question analysis: How watson reads a clue. *IBM J. Res. Dev.*, 56(3):250–263.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020a. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020b. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Infor-*

mation Processing Systems, volume 33, pages 9459–9474. Curran Associates, Inc.

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.

Pedro Rodriguez and Jordan Boyd-Graber. 2021. Evaluation paradigms in question answering. In *Empirical Methods in Natural Language Processing*, page 5.

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

Ellen M. Voorhees. 2019. *The Evolution of Cranfield*, pages 45–69. Springer International Publishing, Cham.

Fengbin Zhu, Wenqiang Lei, Chao Wang, Jianming Zheng, Soujanya Poria, and Tat-Seng Chua. 2021. Retrieving and reading: A comprehensive survey on open-domain question answering.

A Feature List

Feature	Definition
length	The logarithm of the length of the question (number of words).
ablength	If a question is shorter than abnormal length cutoff(default set to 5) this value is set to 1, 0 otherwise.
kincaid	The kincaid readability score (Kincaid et al., 1975) of the question.
duplicates	The sum of the number of duplicate words of the question.
uniques	The number of unique words of the question.
num_nouns	The number of nouns in the question.
num_verbs	The number of verbs in the question.
num_wh	The number of Wh-question words in the question.
num_distinct_wh	The number of distinct Wh-question words in the question.
max_idf	The maximum Inverse Document Frequency (Church and Gale, 1999) of a question.
nq_length_5	If question is shorter than the 5th percentile of the length of NQ, this is -1; 1 otherwise.
nq_length_95	If question is longer than the 95th percentile of the length of NQ, this is -1; 1 otherwise.
no_QB_pattern	If the question has QB pattern like "for 10 points", this is -1; 1 otherwise.
n-gram	The n-gram representation (Kondrak, 2005) of the question.

Table 5: List of features used in NQ-NQ_like Classifier

B Heuristics List

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
clean answer type	Convert "– name this" patterns to "which".	which short story is – name this story by james joyce	which short story is the story by James Joyce
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
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
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 6: List of Heuristics 1.

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 which none is	Convert "which none is" to "which"+answer_type+"is"	which none is in one incident in this book , a man " looked this way and that way"before intervening to stop a killing and hiding the dead body of the murderer in the sand	which second book is in one incident in this book , a man " looked this way and that way"before intervening to stop a killing and hiding the dead body of the murderer in the sand
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 7: List of Heuristics 2.

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 8: List of Heuristics 3.