Addiction Recovery QA: Answering context-specific questions using the ELI5 pipeline

Michael Ayo Dorosan
Asian Institute of Management
Makati City, Philippines
mdorosan@aim.edu

***Abstract*—In this paper, I attempted to investigate the impact of context to the question answering pipeline implemented by the original ELI5 paper by Facebook Research. In contrasting answer quality, I used ROUGE-1, ROUGE-2, and ROUGE-L–where providing supporting documents that is within a question's context allowed for relatively higher ROUGE scores than with an out-of-context support document or with a no-context-at-all input to the answer generator. Additionally, ROUGE scores of answers with the supporting documents were quite low–which possibly signal that abstractive answers do not copy n-gram spans directly from supporting documents. The approach of generating the QA pairs can be greatly improved. However, the analysis followed may serve as a starting point in refining the idea that good LFQA performance can be obtained in a closed domain by simply using within domain supporting documents as inputs to a pre-trained model on an open domain–without the need for further fine-tuning.**

***Index Terms*—question-answering, abstractive generation, document retrieval**

I. Introduction

Question answering (QA) at its core is a task requiring to retrieve information from knowledge. As to where this knowledge is sourced, can vary. To clarify, let's look at three human responses to an interesting question: (i) a response drawn from learned knowledge stored in memory and structured by some form of reasoning, (ii) a response drawn from extracted information stored elsewhere and, lastly, (iii) a response drawn from abstracted–that is, summarized and expressed in original terms–information in memory and elsewhere.

Shifting into the anatomy and physiology of computing machines, information that can be used to formulate answers initially exist as data. And the evolving human ability–combined with technology–to transform this data into information has influenced the techniques by which computers are programmed to succeed in a QA task. One traditional paradigm of transforming data into information for QA is by constructing knowledge bases (often graphs). Consequently, programs are written to traverse such graphs in a manner that mimics human reasoning to arrive at an answer. Such an approach however presents the challenge of organizing vast unstructured data into an informative structure that compositional reasoning can traverse. [1]

Enter the newer paradigm in this QA field–textual question-answering. In contrast to the knowledge base approach, textual question answering leverages on deep neural networks that automates the restructuring of data as well as the reasoning necessary to structure answers from data. Various approaches to this are excellently discussed in a comprehensive paper by Bai et al. [2] This report narrows on one branch of textual QA, that is Long-form question answering (LFQA). Such LFQA task is one that requires more elaborate, free-form answers often containing multiple sentences and in-depth reasoning. To date, the most and perhaps the only LFQA dataset available is the *Explain like I'm Five* dataset [3] which Facebook Research tackles in their LFQA paper back in 2019 [4] where I take the approach I followed in this paper. Figure 1 outlines the general workflow.

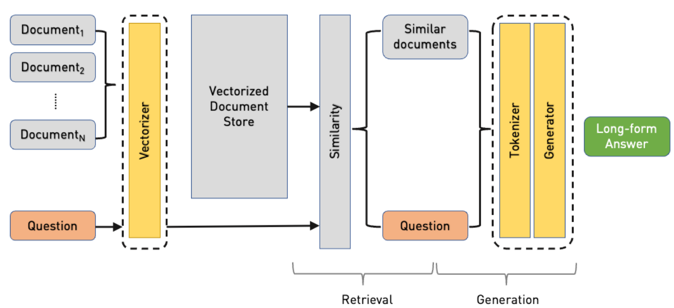

Fig. 1: The general pipeline of abstractive question answering. Stages outlined by dashed lines are transformer models as implemented in this paper. Vectorizers traditionally do sparse transformations to text using TF-IDF, however, the advent of sentence transformers lend themselves for use in transforming text into dense learned representations.

I take a different angle to the point of Fan et al [4] that an abstractive model should essentially compensate for the imperfection or even lack of support document. This I do by raising the question of whether an abstractive model's training introduces biases that overpower the contexts fed by supporting documents in a closed domain. To pursue such a question, I take from a domain which can arguably considered a closed domain–religion. In particular, I look at support documents sourced from the www.churchofchrist.org to simulate the closeness–that is, assuming that religious discourse is heavily biased on doctrine. I further narrow this down to questions that relate to addiction recovery. I also make use of another domain, considered to be more open than that of the prior religious documents–Wikipedia, to generate answers that are supported by out-of-context documents. Moreover, I also explore question answering without any supporting documents and compare these approaches according to the Recall Oriented Understudy for Gisting Evaluation (ROUGE) scores [5].

II. Problem Statement and Set-up

The choice of a using the addiction recovery article search results from www.churchofjesuschrist.org is that religious texts contain strong biases to doctrines. As such, a question asked within these biases will be answered in relation to doctrine. To elaborate further, good quality–or even correct–answers are bounded by religious doctrine.

We then can make comparisons as to whether abstracted answers can be completely biased to favor an inputted supporting documents. Specifically:

1) What is the general workflow of long-form abstractive question answering?
2) How do answers compare in term of ROUGE when varying supporting documents are used: documents within the question's context (addiction recovery, religious text), documents outside of a question's context (Wikipedia snippets), and no supporting documents (question only as input).
3) How much is the word overlap of generated answers with the supporting documents?

A. Data Generation

To obtain closed-domain data, I scraped all the General Conference (GC) and magazine articles returned on the www.churchofchrist.org when the keywords "addiction recovery" is searched. General conferences are semi-annual worldwide meetings where church leaders share inspiring messages drawn from church doctrine while magazine articles are those contained from monthly church publications. An assumption for this dataset is that all documents returned by the search are relevant to the topic of addiction recovery. In total, 10 GC articles and 177 magazine articles were obtained after removing duplicates and non-English texts.

Furthermore, there are two flavors to which scraped texts were restructured into. First, that of an LFQA dataset resembling the ELI5 format. And second, that of a supporting document when later used as context in an abstractive question answering model. For the latter, I sliced documents into 100-word strings. However, for the former since actual question-answer pairs are not available, I followed a QA pair generation approach which closely resembles that of Cloze strategy [6–9]–that is, I parsed texts for sentences likely to be questions by searching for the "?" punctuation and arbitrarily took the 3 sentences that follow as the answer. Such an approach risks catching rhetorical questions–which are often used to introduce a topic in a speech or an article instead of an actual question seeking for a long-form answer. Hence, succeeding sentences may not really constitute a valid QA pair. To mitigate the impacts of such, I manually filtered QA pairs to only contain coherent pairs. There 12 GC and 72 magazine QA-pairs obtained after manual filtering. To ensure no data leakage will occur in such an approach, I only used the 12 GC QA pairs to evaluate the generated answers while using only magazines sourced articles as supporting documents.

B. Supporting Document Indexing

Traditional information retrieval techniques are commonly based on sparse representations of documents in the form of Term-frequency Inverse Document Frequency (TF-IDF) vectors. These vectors are stored in an index for faster retrieval. In contrast to such an approach, is one that makes use of dense vectors which are learned representations obtained from a neural network trained to understand language (i.e., understanding language based on a task such as natural language inference, masked language modeling, etc). Recent scientific excitements has been drawn towards language models that follow an encoder-decoder transformer architecture mainly due to (i) more efficient computation compared to its predecessors and (ii) the concept of attention –which allows a model to focus on features (i.e., in the case of text, words and their positions) relevant to neural network objective [10].

One of the most cited language models of late is BERT–which stands for the bi-directional encoder representations from text. BERT does just that–encode text in a bidirectional manner using a transformer encoder. In the pre-training stage two masked sentences are inputted to the model. BERT is then trained to predict suitable words in-place of masks while simultaneously classifying if the first sentence entails the other. [11]

Since BERT, many have applied fine-tuning of the pre-trained BERT-base (110M parameters) and BERT-large (340M parameters) in many other tasks and have shown record-breaking performance. Of course, performance of such a large model puts downstream deployment constraints into the conversation. Researches dating back to 2006–which at the time was dealing with large ensemble of models–rather than large neural network masked models–propose model compression to address constraints in size, inference time, and the overall computational complexity of large models [12]. Specific to large neural network models, were proposals by Hinton et al back in 2015 [13] which introduced the concept of distillation–a transfer of knowledge in a teacher-student fashion where the teacher is a large high performing model trained on large amounts of data and the student as a smaller (i.e., less parameters) counterpart trained to predict the logits or temperature-scaled softmax output by the teacher for some input. This idea was further developed by Turc et al [14] through their experiments on the initialization of student models–particularly proposing a pre-training as an initialization of student model weights prior to distillation.

Such an approach of pre-trained distillation resulted to smaller student models of comparable performance to larger teacher models. In this paper, I follow a pipeline that make use of `google/bert_uncased_L-8_H-512_A-8`—a transformer variant available via the Hugging Face `transformers` library—as the base language model trained specifically on word-level language understanding tasks. This model is then trained on an ELI5 [4] question-answer sentence entailment task wherein a model learns to contrast a correct question-answer pair with an incorrect match–resulting to a model which can represent a sentence as a dense vector.[15] The training was done by Jernite [16] and the model is also made available via Hugging Face. I then used the resulting retriever model, `yjernite/retribert-base-uncased`, to get dense vector representations for both within-context (i.e., religious, addiction recovery) and outside-context (i.e., Wikipedia) supporting documents. These large collections of short document embeddings were then stored in memory mapped files allowing for reading only a subset of the file into memory from disk instead of reading the entire file into memory.

C. Supporting Document Retrieval

Using the `faiss` [17] library, I then made an index of the memory mapped files that is available either in GPU (for smaller within-context embeddings) and CPU (for the outside-context Wikipedia embeddings that do not fit in GPU). The index allows for faster supporting document retrieval which is qualified by the selecting only the most similar *k* documents to a question using the maximum inner product as similarity metric. I observed that while retrieval times are faster for GPU-stored indexes, the difference is not outstanding for my question-answering task.

D. Answer Generation

In this paper, I follow the abstractive answer generation–that is, given a question and a supporting document as inputs, an answer is generated using a text generation model. Let me break that down into components. First, I leverage on an existing language model that is a variant of denoising autoencoders (e.g., BERT) which are trained to reconstruct a masked text input. These models are usually pre-trained on large amounts of text and are readily available in the Hugging Face library. Here, my model of choice is BART which builds on BERT variants by adding more novel corruptions on input text. These model corruptions are noising strategies which, like in masked language modeling, should allow a transformer autoencoder to successfully reconstruct the denoised text. The proponents of BART [18] posit that the combined BERT-encoder and GPT-decoder [19] architecture implemented allow for a broader choice of noising functions. Aside from token masking, BART was pre-trained on reconstructing input that was subject to (i) sentence permutation, (ii) document rotation, (iii) token deletion, and (iv) the then novel text infilling noising functions.

Now taking this BART model–particularly, BART large with 10 percent more parameters than its BERT counterpart–Jernite [16] then fine-tuned it to a contextualized long-form question answering task using the ELI5 [4]. The inputs are (i) a question from the ELI5 training set, and the (ii) concatenated top-*k* similar Wikipedia documents snippets to the question (*k=10*). A question can be paired with a different answer that also answers the question. This is made possible due to having more than one good (i.e., highly rated) answer available from the dataset. The resulting fine-tuned model, `yjernite/bart_eli5`, is used as the abstractive answer generator.

III. Experiments

The following sections discusses the experiments conducted to investigate on the viability of contextualizing an open-domain question-answering pipeline using supporting documents from a closed domain–speeches, stories, and articles specific to a religion and topic.

A. Answer Evaluation

To facilitate the comparison, I examine the ROUGE scores if different text pairs. ROUGE measures the similarity between a computer-generated text with one or more reference texts. Commonly used metrics are ROUGE-N variants, which account for the ratio of N-gram overlaps between a candidate text and a reference text, see (1).

$$\text{ROUGE-N} = \frac{\sum_{S \in \text{Ref}} \sum_{\text{n-gram} \in S} |\text{n-gram}_{match}|}{\sum_{S \in \text{Ref}} \sum_{\text{n-gram} \in S} |\text{n-gram}|} \tag{1}$$

In 1, I show the recall variant of ROUGE-n where the count of matching N-grams is normalized by the total number of N-grams present in reference texts. Another ROUGE variant is the ROUGE Longest Common Sequence or ROUGE-L, which already includes within-sequence N-grams–where N can be any integer value. This is so because the longest common sequence of two texts X and Y, $LCS(X,Y)$, is defined as the sequence length k where strictly the elements of X and Z are equal, $x_{i...k} = z_{j...k}$. Variants of ROUGE-L vary on the normalizing factor, n, in the ratio $LCS(X,Y)/$n. Choices can be either the length of the reference (ROUGE-L recall) or the length of the candidate sequence (ROUGE-L precision). Moreover, since precision and recall can be calculated, they can also be weighted according to some β to get the F_β measure. In the succeeding sections ROUGE 1-gram, 2-gram, and LCS are used for comparison.[5]

TABLE 1: 1-gram ROUGE scores comparison for supporting document inputs sourced within the domain, from a more general collection (out-of-domain, Wikipedia articles), as well as a generator input with no supporting documents (no context)

ROUGE 1-gram	Within-domain	Out-of-domain	No Context
Precision	**0.1971**	0.1786	0.1011
Recall	**0.2262**	0.1989	0.2111
F1-score	**0.2024**	0.1805	0.1352

TABLE 2: 2-gram ROUGE scores comparison for supporting document inputs sourced within the domain, from a more general collection (out-of-domain, Wikipedia articles), as well as a generator input with no supporting documents (no context)

ROUGE 2-gram	Within-domain	Out-of-domain	No Context
Precision	**0.0259**	0.0145	0.0057
Recall	**0.0322**	0.0155	0.0115
F1-score	**0.0274**	0.0146	0.0074

TABLE 3: Longest common sequence ROUGE scores comparison for supporting document inputs sourced within the domain, from a more general collection (out-of-domain, Wikipedia articles), as well as a generator input with no supporting documents (no context)

ROUGE LCS	Within-domain	Out-of-domain	No Context
Precision	**0.1192**	0.1090	0.0708
Recall	0.1392	0.1254	**0.1472**
F1-score	**0.1224**	0.1112	0.0944

Results show that supporting documents within the domain of the question context achieve higher rouge scores with the reference answers. This affirms that despite the model being trained on large out-of-domain datasets (i.e., Wikipedia, and ELI5), biases on the training data are dampened by the use of within-domain supporting documents–even without any fine-tuning on the addiction recovery religious texts.

```
Within context supporting documents. Magazine articles from churchofchrist.org

Question: How will going to church help me quit my addiction?

Generated Answer:  It won't. Going to church isn't going to help you quit your addiction. It's going to make you
feel like you're doing something that you're not supposed to be doing. The only way to quit an addiction is to s
top doing the thing you're addicted to. If you don't want to do it, don't do it.


Out-of-context supporting documents. (Wikipedia articles)

Question: How will going to church help me quit my addiction?

Generated Answer:  Going to church doesn't help you quit your addiction. Going to AA meetings is a way to help p
eople quit their addiction. It doesn't work for everyone, but it does work for a lot of people. If you want to q
uit, you need to find someone who is willing to help you with your problem.
```

Fig. 2: Sample answers generated using within-domain and out-of-domain support

Figure 2 show that out-of-domain supporting document surfaced a well-known addiction recovery/rehabilitation program (i.e., AA), while within-domain supported answer highlighted the church to cause someone to "feel like you're doing something that you're not supposed to be doing" which may be a more likely answer when consulting a church representative.

B. Answer-Context Overlap

In this section, we verify the abstractive ability of the generator. Specifically, by the distribution of ROUGE recall scores of the generated and actual (reference) answer, against the supporting document snippets used as context. For each top k references used ($k = 10$), ROUGE scores with the generated and actual answers were calculated and the maximum ROUGE observed for each group was recorded. This was done for all 12 question-answer pairs in the test set. The distributions of max pairwise ROUGE recall are shown in Figure 3.

We can observe that the max ROUGE recall distribution of generated abstractive answers had less variance, indicating the consistency of the abstractive capability of the generator across 12 QA pairs in the evaluation set–that is, abstraction may have accounted for all the supporting documents.

IV. Conclusion and Recommendations

Within context supporting documents resulted to higher ROUGE answers when compared with the reference answers.

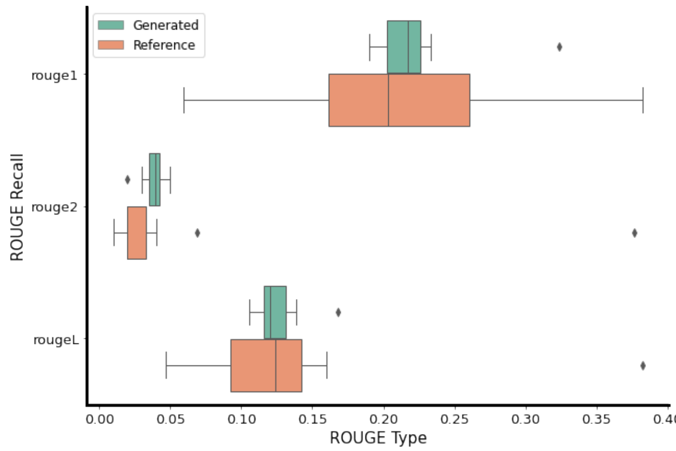

Fig. 3: Overlap between a supporting document as context reference and the generated answer is shown to probe how similar the answer is to each of the references.

The approach of generating the QA pairs can be greatly improved. However, the analysis followed may serve as a starting point in refining the idea that good LFQA performance can be obtained in a closed domain by simply using within domain supporting documents as inputs to a pre-trained model on an open domain–without the need for further fine-tuning.

References

[1] Das, R., Zaheer, M., Reddy, S., & McCallum, A. (2017, July). Question Answering on Knowledge Bases and Text using Universal Schema and Memory Networks. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers) (pp. 358-365).

[2] Bai, Y., & Wang, D. Z. (2021). More Than Reading Comprehension: A Survey on Datasets and Metrics of Textual Question Answering. arXiv preprint arXiv:2109.12264.

[3] https://github.com/facebookresearch/ELI5

[4] Fan, A., Jernite, Y., Perez, E., Grangier, D., Weston, J., & Auli, M. (2019, July). ELI5: Long Form Question Answering. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (pp. 3558-3567).

[5] Lin, C. Y. (2004, July). Rouge: A package for automatic evaluation of summaries. In Text summarization branches out (pp. 74-81).

[6] Hermann, K. M., Kocisky, T., Grefenstette, E., Espeholt, L., Kay, W., Suleyman, M., & Blunsom, P. (2015). Teaching machines to read and comprehend. Advances in neural information processing systems, 28.

[7] Cui, Y., Liu, T., Chen, Z., Wang, S., & Hu, G. (2016). Consensus attention-based neural networks for Chinese reading comprehension. arXiv preprint arXiv:1607.02250.

[8] Hill, F., Bordes, A., Chopra, S., & Weston, J. (2015). The goldilocks principle. arXiv preprint arXiv:1511.02301.

[9] Bajgar, O., Kadlec, R., & Kleindienst, J. (2016). Embracing data abundance: Booktest dataset for reading comprehension. arXiv preprint arXiv:1610.00956.

[10] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. Advances in neural information processing systems, 30.

[11] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

[12] Buciluǎ, C., Caruana, R., & Niculescu-Mizil, A. (2006, August). Model compression. In Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 535-541).

[13] Hinton, G., Vinyals, O., & Dean, J. (2015). Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531, 2(7).

[14] Turc, I., Chang, M. W., Lee, K., & Toutanova, K. (2019). Well-read students learn better: On the importance of pre-training compact models. arXiv preprint arXiv:1908.08962.

[15] Reimers, N., & Gurevych, I. (2019). Sentence-bert: Sentence embeddings using siamese bert-networks. arXiv preprint arXiv:1908.10084.

[16] https://github.com/huggingface/notebooks/

[17] https://github.com/facebookresearch/faiss

[18] Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., ... & Zettlemoyer, L. (2019). Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. arXiv preprint arXiv:1910.13461.

[19] Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., ... & Amodei, D. (2020). Language models are few-shot learners. Advances in neural information processing systems, 33, 1877-1901.