Statistical Natural Language Processing
Literature Review

Introduction

Standardised tests have been long an essential part of the admission process to various educational institutions. They measure essential and nontrivial skills and determine whether they align well with the requirements. To that end, the development of test comes with the price of expensive intellectual labours and time-consuming tasks, not to mention possible bias.

We intend to simplify the process by using recent research within deep learning. Our goal is to build a system that generates multiple choice question-answer pairs based on the provided paragraphs.

Question-Answer Generation

In the following section, we provide an overview of methods for Question-Answer Generation. We mainly focus on two types of models: RNNs and Tranformers.

RNN-based models

We started our exploration from RNN based models. Most of the recent paper follow more or less similar methodology: use both,a paragraph and an answer, as an input for question generation. (Zhao et al. 2018) argue that usage of *whole* paragraphs offers a more comprehensive context. Their model is based on Seq2Seq architecture (i.e. LSTM). To ensure the generation of questions that are *meaningfully*-aligned with the answer, the authors employ **answer tagging technique**. It supplies one-hot indicators of the position of answer span in the context-passage during the encoding phase. Additionally, the authors address long sequence processing by proposing the **gated self-attention mechanism** in the encoder (Zhao et al. 2018) (Vaswani et al. 2017). This type of attention implements a learnable vector that decides how much information should be kept from the processed sequence (e.g. the one that passed through the self-attention) and how much from the original one. It effectively fuses information and provides a more robust encoding of passage-answer representation. At the same time, authors implement **copy** and **maxout pointer** mechanisms (Zhao et al. 2018). The copy mechanism allows to copy words from source to target sequence, it is a useful technique to compensate for the unknown words (Gulcehre et al. 2016). However, copying causes repetition of tokens on the output sequence. Authors suggest to use maxout pointer layer to eliminate the issue [1].

In (Zhou et al. 2017) authors use Seq2Seq model (i.e. bidirectional GRU) to generate open-ended questions. Similar to the previous method, the model takes in a paragraph, an answer position indicator. Meanwhile, the model also processes lexical features, such as word case, NER and POS. Authors also use **pointing mechanism** proposed by (Gulcehre et al. 2016) to solve **rare and unknown words problem** (Zhou et al. 2017). For example, *Ceasar* in the English sentence (source) is written identically in the French sentence (target). Thus, if network has never seen *Ceasar* before, the pointing mechanism will tell network to simply copy this word into the target sentence. The final solution is able to outperform rule-based and vanilla Seq2Seq models with attention (Zhou et al. 2017).

(Willis et al. 2019) suggest a method to eliminate the need of a correct answer. The proposed method is based on the idea of **Key Phrase Extraction** (KPE): given a paragraph model should identify most *interesting* terms, which then might be used as input-answers along with the paragraph. (Willis et al. 2019) employ encoder-decoder with bidirectional LSTM module. As an input, the model takes GLoVE embeddings (Pennington, Socher, and Manning 2014), part-of-speech tags and named-entities tags. The concatenated input passes through the encoder with the self-attention layer (Vaswani et al. 2017), that is followed by LSTM decoder. Generated *key phrases* are evaluated on the ground-truth answers from SQUAD. The authors showed that KPE model outperforms baseline model that chooses Named Entities as key-phrases. It is also shown that KPE generated answers together with **QG-Net** provide question-answer pairs that closely resemble those of the human-experts (Willis et al. 2019).

Next paper (Liu et al. 2020) makes one steps further and proposes end-to-end system for question-answer generation, which they call **Action-Clue-Style aware** approach. The method focuses on three aspects: *answer extraction*, *clue ex-*

[1] we do not describe the mechanism as it is too advanced in the context of this project

traction (part of the paragraph that contains the information for the question) and *style* (which user can specify to generate specific type of question, e.g *how*, *what*, *yes-no* etc). Authors implement two models: **Seq2Seq** and **Transformer based** [2] (Liu et al. 2020). Seq2Seq model consists of GRU-based encoder-decoder. The encoder takes a whole paragraph that is represented with word embeddings, named entity indicators, part-of-speech indicators, span of the answer indicator (optional), and span of the question (optional). The decoder uses the output of encoder, as well as style-indicator to generate question-answer pairs. The **Transformer**-based model uses similar input, along with the position and segment embedding.

An interesting addition to the ACS is **filtering** (Liu et al. 2020). Resulting question-answer pairs (both, from Seq2Seq and Transformer) go through BERT-like model that evaluates the quality of the result and *filters out* weak pairs. While Answer-Clue-Style model outperforms many state-of-the-art models (such as BERT-QG-QAP, NQG-LM) (Liu et al. 2020), it has a major drawback. To find spans for answer, clue and question (in the case, it is not provided), the model divides paragraphs into multiple chunks, where each chunk is evaluated (for a *fitness*) based on the conditional probability. For example, span of the answer (i.e. chunk that is chosen as a span for the answer) depends solely on POS, NER and length of the chunk (Liu et al. 2020).

In (Kumar et al. 2018), the authors propose similar method to *ACS*. Main difference is that instead of the conditional probabilities of chunks, the model evaluates the whole paragraph. So to find the *pivotal answers* (i.e. model uses a unidirectional LSTM encoder that is searching for named entities in the paragraph. The most probable named-entities become the answer-spans. The authors does not provide the comparison of results with state-of-the-art models.

Transformer-based methods

Even thought RNN-based methods provide a good performance that is often comparable (or even superior) to the level of human-experts, we can see that Transformer based model provide much better results.

For example, in (Chan and Fan 2019), authors use pretrained BERT and finetune it for the QA task. The model takes the context paragraph and the correct answer as an input. The last layer is then decoded into the question. To improve the generation of the questions, authors modified the decoding procedure (Chan and Fan 2019) by introducing the **sequential decoding**. In *vanilla* BERT, each output is decoded independently from other tokens. However, this model takes into account how previous tokens in the sequence are decoded, to decode the current token. Authors also observed that the model struggled with long sequences, as well as ambiguity if the answer (i.e. words that are part of the answer) appear multiple times in the input paragraph (Chan and Fan 2019). To address the issue, the authors use the special token **[HL]** to indicate the correct location of the answer in the paragraph. Together BERT with two modifications achieves state-of-the-art performance on SQUAD dataset.

Most QA works focuses on open-question-answer generation, while our interest is mainly focused on the multiple-choice type of questions. One of the works that explores this area is (Lelkes, Tran, and Yu 2021). The proposed method consists of two stages: *quiz-style question-answer generation* and *distractor generation*. For a quiz-style question-answer generation step, the goal is to generate questions and correct answers (given the input paragraph). For distractor generation step, the model generates incorrect alternative answers that are both plausible solutions to the question and distinct from the correct answer. The distractor does so based on the input paragraph, generated question, and a generated correct answer. The authors use two transformer-based models: pre-trained PEGASUS and T5 models. The evaluation and finetuning is performed on NewsQuizQA. The models outperform strong baselines (Lelkes, Tran, and Yu 2021) (such as *vanilla* PEGASUS and T5 fine-tuned for QA tasks).

Generating the *incorrect* answers seems to be a difficult task (Lelkes, Tran, and Yu 2021). Work by (Gao et al. 2019) also focuses on **distractor answer** generation. Authors employed LSTM model to generate a set of distractors given the paragraph, the question, and the corresponding correct answer. The model consists of the **hierarchical encoder** with the **static attention** mechanism that determines the importance of words and sentences. It is followed by the **gated attention** that saves the information related to the questions. The decoder then takes the representation of words and sentences along with the embedded question to generate the distractors. Authors claim that it provides grammatically and contextually consistent answers (Gao et al. 2019).

Summary

Our review shows that multiple-choice question generation is an emerging area of the NLP research. At the same time, there are no many works that focus purely on the examination data.

Our project is to focus focus on the Transformer-based solutions. We will draw inspiration from (Chan and Fan 2019). We will use **pretrained BERT to encode** paragraphs and correct answers. The idea is that the person who wants to generate question answer pair would input the article/paragraph and mark some part of the article/paragraph as an answer. We would then utilise **sequential decoding** for question generation. We would then use the **distractor generator** from (Lelkes, Tran, and Yu 2021) to generate incorrect answers.

If time allows, we will work on the **question filtering** by implementing the algorithm from (Liu et al. 2020).

We will use RACE (Lai et al. 2017) and, possibly, SciQ Datasets[3] for training. We will evaluate a model based on the test subset from RACE dataset. Our evaluation would include the measure of Perplexity, BLEU score, ROUGE-L score, and METEOR score for questions and distractor answers. We will compare our results to (Jia et al. 2020).

[2] while it is not related to RNNs, we describe transformer-based model to keep an consistent flow of the report

[3] https://allenai.org/data/sciq

References

Chan, Y.-H., and Fan, Y.-C. 2019. A recurrent bert-based model for question generation. In *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*, 154–162.

Gao, Y.; Bing, L.; Li, P.; King, I.; and Lyu, M. R. 2019. Generating distractors for reading comprehension questions from real examinations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 6423–6430.

Gulcehre, C.; Ahn, S.; Nallapati, R.; Zhou, B.; and Bengio, Y. 2016. Pointing the unknown words.

Jia, X.; Zhou, W.; Sun, X.; and Wu, Y. 2020. Eqg-race: Examination-type question generation. *arXiv preprint arXiv:2012.06106*.

Kumar, V.; Boorla, K.; Meena, Y.; Ramakrishnan, G.; and Li, Y.-F. 2018. Automating reading comprehension by generating question and answer pairs.

Lai, G.; Xie, Q.; Liu, H.; Yang, Y.; and Hovy, E. 2017. RACE: Large-scale ReAding comprehension dataset from examinations. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 785–794. Copenhagen, Denmark: Association for Computational Linguistics.

Lelkes, A. D.; Tran, V. Q.; and Yu, C. 2021. Quiz-style question generation for news stories. *arXiv preprint arXiv:2102.09094*.

Liu, B.; Wei, H.; Niu, D.; Chen, H.; and He, Y. 2020. Asking questions the human way: Scalable question-answer generation from text corpus. In *Proceedings of The Web Conference 2020*, 2032–2043.

Pennington, J.; Socher, R.; and Manning, C. D. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 1532–1543.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention is all you need. *arXiv preprint arXiv:1706.03762*.

Willis, A.; Davis, G.; Ruan, S.; Manoharan, L.; Landay, J.; and Brunskill, E. 2019. Key phrase extraction for generating educational question-answer pairs. In *Proceedings of the Sixth (2019) ACM Conference on Learning@ Scale*, 1–10.

Zhao, Y.; Ni, X.; Ding, Y.; and Ke, Q. 2018. Paragraph-level neural question generation with maxout pointer and gated self-attention networks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 3901–3910.

Zhou, Q.; Yang, N.; Wei, F.; Tan, C.; Bao, H.; and Zhou, M. 2017. Neural question generation from text: A preliminary study.