Università di Pisa

Human Language Techonology

Project Report

Filippo Biondi
f.biondi12@studenti.unipi.it
Roll number: 596789

Paul M. Magos
p.magos@studenti.unipi.it
Roll number: 588669

Matteo Tolloso
m.tolloso@studenti.unipi.it
Roll number: 598067

November 19, 2023

Contents

1 Introduction

Question generation is the task of automatically creating questions from a given text, such as a passage, a sentence, or an answer. Question generation can be useful for educational purposes, such as creating practice questions, enhancing reading comprehension, and facilitating self-assessment. Question generation can also be helpful for other natural language processing tasks, such as question answering, dialogue systems, and summarization.

Question generation is a challenging problem that involves various aspects, such as linguistic analysis, semantic understanding, logical reasoning, and natural language generation. Depending on the input and the output format, question generation can be classified into different types, such as:

- Answer-aware question generation: given a passage and an answer (or an answer span), generate a question that is specific to the answer.
- Answer-agnostic question generation: given a passage, generate questions that can be answered from the passage.
- Conversational question generation: given a dialogue context, generate a question that can continue the conversation.
- Distractor generation: given a question and a correct answer, generate incorrect answers (distractors) that are plausible and relevant.

In this project we are going to focus on the first two task analyzing different models and comparing them according to some selected metrics that will be discussed in the following.

2 Background

NLP techniques involve a combination of linguistics, computer science, and machine learning, aiming to bridge the gap between human communication and computer understanding. Early NLP efforts were rule-based and

relied heavily on linguistic rules and grammatical structures. However, recent advancements have shifted the focus to data-driven approaches, leveraging large datasets and deep learning models for improved performance in various language-related tasks.

NLP encompasses a wide range of applications, including but not limited to sentiment analysis, machine translation, text summarization, and question generation. Key challenges in NLP include resolving ambiguities, handling syntactic and semantic complexities, and capturing context-specific information. Various mathematical and computational models are employed in NLP, such as probabilistic models, language models, and deep learning architectures, which have significantly enhanced the performance of NLP systems.

Notable milestones in the field of NLP include the development of statistical language models, such as the n-gram models [1], and the introduction of recurrent neural networks (RNNs) for language modeling [13]. More recently, the introduction of transformer-based architectures, such as the attention mechanism [18], has revolutionized the NLP landscape, leading to significant improvements in tasks like machine translation and natural language understandinsg.

2.1 Datasets

This subsection describes the datasets that we used to finetune each of our models.

2.1.1 SQuAD dataset

The Stanford Question Answering Dataset (SQuAD) is a reading comprehension dataset that consists of questions posed by crowdworkers on a set of Wikipedia articles. The dataset is designed to test the ability of a system to answer reading comprehension questions and to abstain from answering when presented with a question that cannot be answered. Each question in the dataset has an associated context paragraph, which is a segment of text from the corresponding Wikipedia article. The answer to every question is a segment of text, or span, from the corresponding

reading passage, or the question might be unanswerable.

2.1.2 SQuAD italian

SQuAD-it is derived from the SQuAD dataset and it is obtained through semi-automatic translation of the SQuAD dataset into Italian. It represents a large-scale dataset for open question answering processes on factoid questions in Italian. The dataset contains more than 60,000 question/answer pairs derived from the original English dataset.

2.2 Models

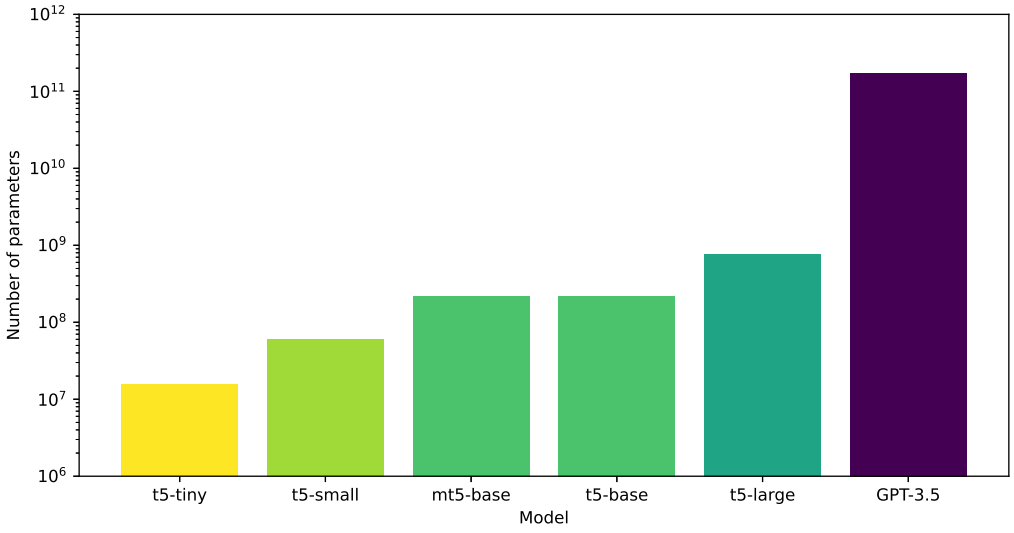

Figure 1: Models parameters cardinality

We are going to introduce the models that we used in our experiments. In particular here we summarize the architecture and the training framework of the models in their original paper, while in section 3.2 we describe how we used each of these models in our scenario.

2.2.1 T5 architecture

The T5 model [17] is a text-to-text transformer-based model that can handle any NLP task by converting both the input and output to natural language. It uses an encoder-decoder architecture with pre-trained parameters from a large unstructured text corpus and a multi-task mixture of supervised tasks. The model has several variants with different sizes, ranging from 220 million to 11 billion parameters. Larger models tend to perform better on downstream tasks. From the Figure 1 you can have an idea of the size of the different t5 models together with other models that we used.

The T5 models were trained using a technique called transfer learning, where a model is first pre-trained on a data-rich task before being fine-tuned on a downstream task. The pre-training includes both supervised and self-supervised training. Supervised training is conducted on downstream tasks provided by the GLUE and SuperGLUE benchmarks, while self-supervised training uses corrupted tokens by randomly removing 15% of the tokens and replacing them with individual sentinel tokens. The input of the encoder is the corrupted sentence, the input of the decoder is the original sentence, and the target is then the dropped-out tokens delimited by their sentinel tokens.

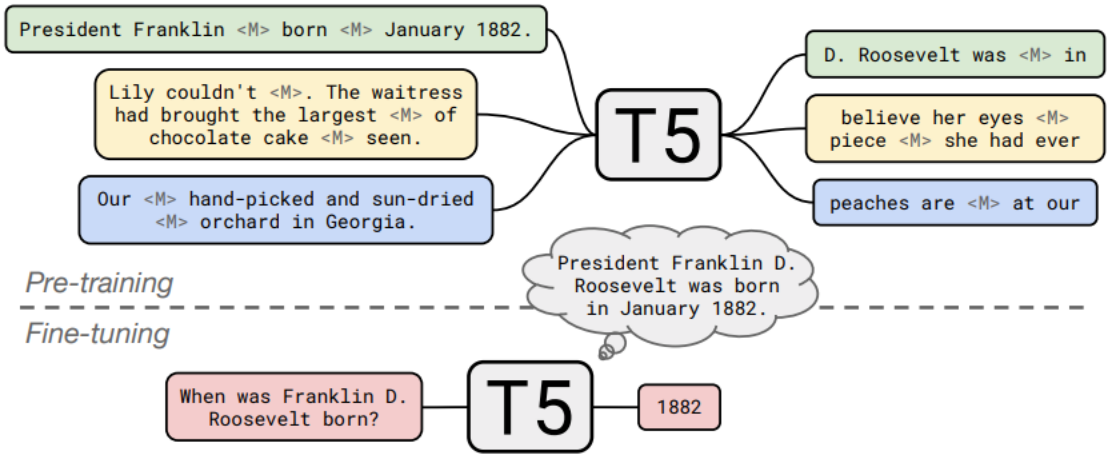

Figure 2: Caption

3 Methods

In this section, we present a comprehensive account of the methodologies employed in our experimental study. The methodology is organized into several key components.

Firstly, we delve into the pre-processing phase, where we exploit the dataset handling procedures and tokenization strategies applied. Then, we introduce the models we used to accomplish ours goals explaining our choices and intents. Conclusively, our attention turns to the evaluation methodology that guided our assessment of results. The following sections will expound upon each of these components, offering an in-depth understanding of our research procedures.

3.1 Preprocessing

In order to perform answer-aware question answering we need a way to indicate to the model which is the answer inside the context. To do so we experimented two main strategies. The first one consists in surrounding

the answer in the context with special tokens. The second one consist in repeating the answer at the beginning of the context separating them with a special token (similarly to what is done in question answering models for the question).

After some preliminary test we didn't noticed major differences between the results of this two approaches and thus we decided to use the second one since it allow us to use answers that are not necessarily contained in the context. Furthermore in this way we can ensure that the answer is never missing due to truncation of the context during preprocessing (as can be seen from the code we used a maximum length for the context of 256 tokens).

```
1  class MyDataset_question_generation(Dataset):
2
3      def __init__(self, data, tokenizer):
4          self.data = data
5          self.tokenizer = tokenizer
6
7      def __getitem__(self, index):
8          context = self.data[index]["context"]
9          new_context = self.data[index]["answers"]["text"][0] + "</s>" +
               ↪    context
10
11         tokenized_in = self.tokenizer(
12             new_context,
13             text_target=self.data[index]["question"],
14             max_length=256,
15             truncation="only_first",
16             padding="max_length",
17             return_tensors="pt",
18         )
19         tokenized_in["labels"][tokenized_in["labels"] == self.tokenizer
               ↪ .pad_token_id] = -100
20         return {"input_ids": tokenized_in["input_ids"][0],  "labels":
               ↪ tokenized_in["labels"][0],
21                 "attention_mask": tokenized_in["attention_mask"][0]}
22
23     def __len__(self):
24         return len(self.data)
```

Code 1: Preprocessing of items

In the Code 1 show the creation of a subclass of the `PyTorch` class "`Dataset`". In our implementation, when a new item is requested the new context is created concatenating the answer with the old context (line 9).

From the call to the tokenizer (line 11) you can see that we decided to truncate the context to 256 tokens. Since this snippet is for the question generation task, obviously the training target will be the question.

3.2 Models and experiments

We first focused on the problem of answer aware question generation, in which given the context and the answer the model had to generate a suitable question that is answered by the provided answer (which should be present in the context). To address this task we decided to employ a sequence to sequence model, in particular we finetuned some pretrained `T5` models. Also we trained some `mT5` which are pretrained model on the multilingual *mc4* dataset (104 languages). Once obtained a functioning model for answer-aware question generation our next challenge was answer-agnostic question generation. We decided to exploit the already trained models to preform question generation using an answer generated by another model, dividing the main task in two sub-task (one of which was already solved by the previously trained T5 models). To perform the other task, answer generation given the context only, we used again some *T5* models. In this way the answer could be also not present in the context (even if it must be said that since the model is fine tuned with answer that are always part of the context it is very unlikely that the generated answer is not present in the context). To simplify the composition of the two different kind of models we used only the T5 models for question generation trained with the answer at the beginning of the context. In addition we wanted to test if a single T5 model alone was able to perform the whole task, so we trained it as an end-to-end model in order to generate a question answer pair starting from the context only. Finally we experimented zero shot question generation (both answer-aware and answer-agnostic) using OpenAI ChatGPT-3.5 to see if a much more bigger model with respect to those tested in our experiments, even if not finetuned for this specific task, would outperform ours models.

In the following listing, you can find a summary of the model and the experiments that we performed.

- t5 small/base/large for question generation using special tokens to

enclose the answer in the context

- t5 tiny/small/base for question generation with the answer inserted at the beginning of the context. In this way the answer could also be not in the text and could be generated by another model
- mt5 base for question generation with the answer inserted at the beginning of the context on SQuAD Italian.
- t5 small/base for answer generation
- t5 small/base for end-to-end answer agnostic question generation
- combination in pipeline of t5 base for answer generation and t5 small for question generation to perform answer-agnostic question generation.
- zero shot prompting with Chat-GPT-3.5 for question generation (both answer-aware and answer-agnostic)

3.3 Evaluation criteria

Evaluating the result of a question generation model is not an easy task. Common metric to compare the generated question with the reference question contained in the dataset, such as BLEU [15] or ROUGE [11], are based on n-gram analysis, so they just measure a syntactical difference between the two questions. Instead a good metric for question answering task should grasp the meaning of the generated question and check if the question is answered by the provided answer in case of answer-aware question generation, or if the generated question answer pair is coherent and relative to the context in case of answer-agnostic question generation. To reach this goal we looked in the litterature to find metrics that meet this requirements. We found two promising metrics, RQUGE [14] and QAScore [7] that are able to calculate a score given the context, a reference answer and the generated question. This metrics were suitable for assessing our answer-aware question generation models while for the answer-agnostic we found in litterature QRel [19], a metric that only need the context and

the generated question, without requiring any answer. In addition the previous two metric (RQUGE and QAScore) are still usable for assesing answer-agnostic results if used with the generated answer instead of the reference answer.

3.3.1 RQUGE

RQUGE (Reference-free QUestion Generation Evaluation) is a metric that can compute the quality of the candidate question without requiring a reference question. As we can see in Figure 3 this metric employ a question answering model to generate an answer given the generated question. Then the such obtained answer is passed to another model that will assign a score to the quality of the generated answer with respect to the reference answer (taking into account also the context and the question). In this way if the generated question is different (both syntactically and semantically) from the reference question but the answer to that question is the right one the metric will assign an high score. This allow us to evaluate better a model and avoid the models overfitting the dataset. The model used for question generation is UnifiedQAv2 [8] which is a T5-based encoder-decoder model trained on 20 QA datasets. For the span scorer model is used an encoder-only BERT-based model fine-tuned on MOCHA [2], a dataset of human judgment scores, where annotators were asked to score candidate spans, given the context, gold answer, and the corresponding question.

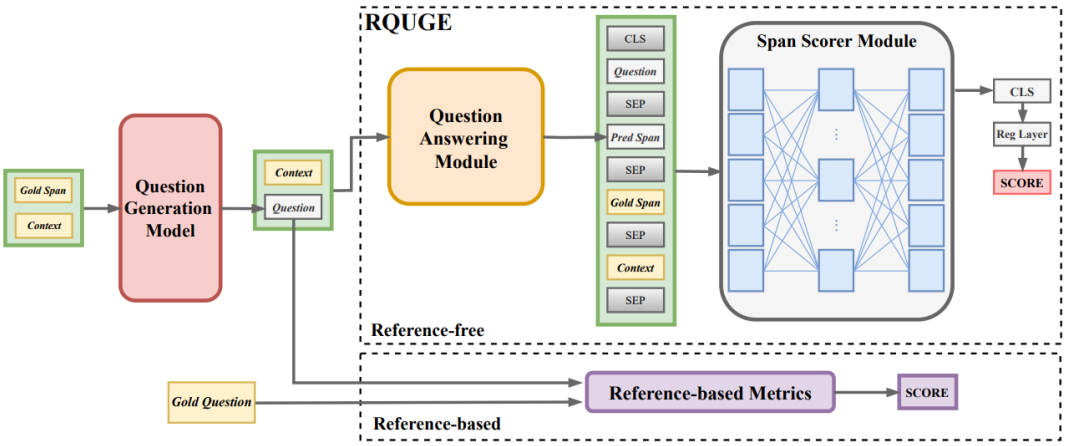

Figure 3: Diagram of RQUGE calculation

3.3.2 QAScore

QAScore is an automatic metric for evaluating question generation systems, which is unsupervised and reference-free. QAScore uses a pretrained language model, RoBERTa [12], to measure the cross entropy of predicting the masked words in the reference answer relative, given the context and the generated question. Like RQUGE, QAScore can evaluate a question without comparing it to a human-generated reference. And in the paper where it is introduced the authors show that it correlates better with human judgments than existing metrics such as BLEU, METEOR, BERTScore and BLEURT, according to a crowd-sourced human evaluation experiment on the HotpotQA dataset.

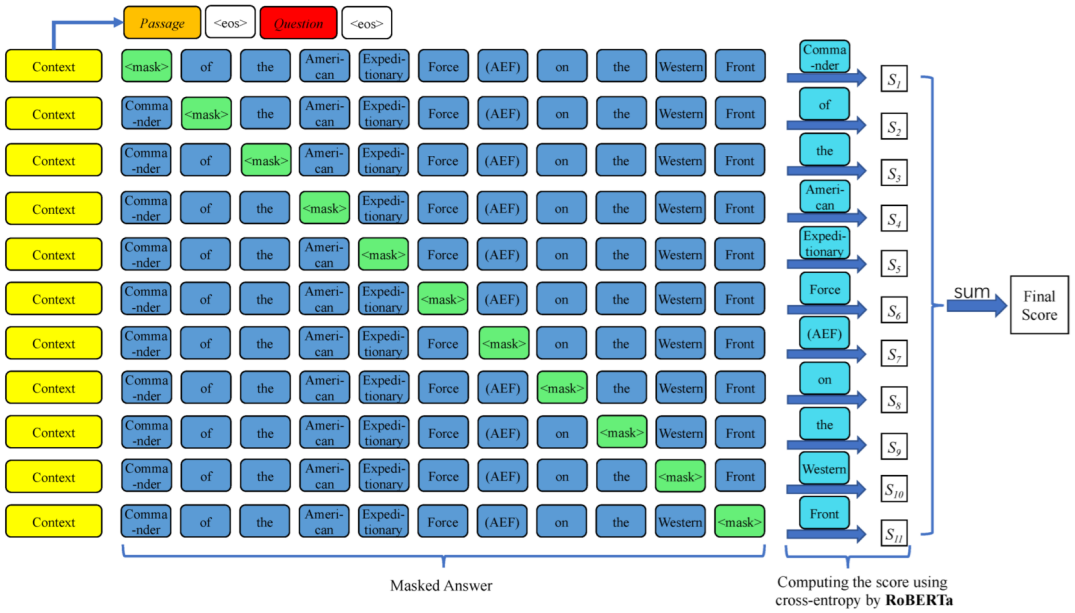

Figure 4: Explanation of how QAScore is calculated

3.3.3 QRel

QRelScore is a context-aware relevance evaluation metric for question generation. It aims to address the weaknesses of existing metrics that do not consider the input context of generation and may wrongly penalize legitimate and reasonable candidate questions. QRelScore consists of two scoring components: local relevance matching (QRelLRM) and global relevance generation (QRelGRG). QRelLRM uses layer-wise embeddings and cross attention from BERT [3] to capture the word-level similarity between the candidate and context. QRelGRG uses GPT2 [16] to measure the factual consistency between the candidate and all possible evidences in the context by comparing the confidence of generating the context with or without the candidate as a prompt. It achieves higher correlation with human judgments than existing metrics on three dimensions: grammaticality, answerability, and relevance. It also shows better robustness and efficiency in detecting adversarial samples and improving question answering performance.

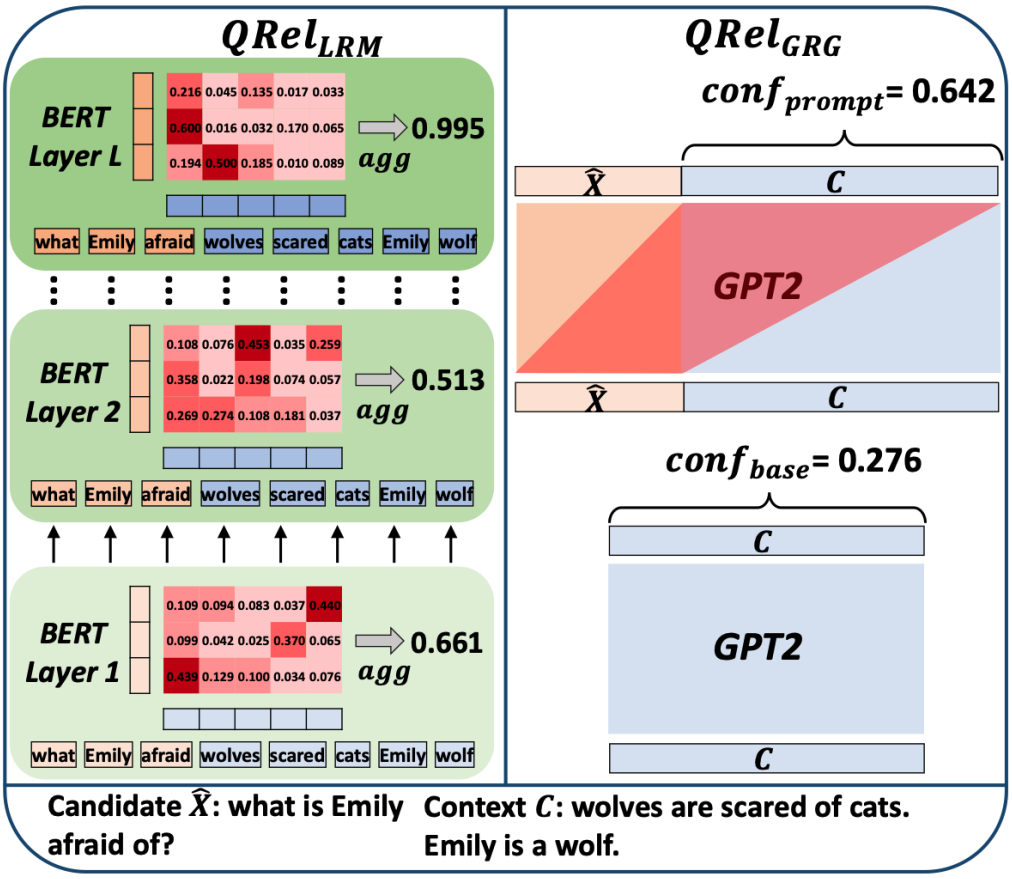

Figure 5: Explanation of the two model that compose QRel

4 Results

In this section, we present a comparative analysis of the performance of the models referenced in Section 3.2. The metrics utilized for evaluation include BLEU, ROUGE, RQUGE, QAScore, and QRel. To address ambiguity regarding the score range and interpretation of certain metrics, we opted to apply them to the questions and answers within the dataset. The resultant values were then treated as the "ground truth" for our assessment. Specifically, the QAScore exhibits a negative value, and our empirical assessments indicate a convergence toward zero when the input quality is high.

The QRel score, designed to evaluate the alignment of generated ques-

tions and answers within an end-to-end context, appears to be influenced by the challenge of handling long contexts, despite the original paper's assertion that the QRel metric performed well on the SQuAD dataset.

Interestingly, the scores achieved by our models surpass the specified maximum. This discrepancy could be attributed to the observed difficulty in accurately capturing and aligning information within lengthy contexts, leading to an elevated QRel score. While the original paper may have demonstrated success, our findings underscore the importance of considering the impact of context length, revealing a potential limitation in the applicability of QRel to scenarios involving extensive textual information.

4.1 Question Generation

The evaluation on the Squad dataset reveals that T5 models, particularly T5-large, consistently outshine other models across multiple metrics such as BLEU, ROUGE1, ROUGE2 and RQUGE. This underscores the significance of model size, with larger T5 variants demonstrating superior capabilities in generating meaningful questions. In comparison, chatGPT-3.5 performs reasonably well, but it falls behind T5-large in most aspects.

Model	**Dataset**	**Score**				
		BLEU	**ROUGE1**	**ROUGE2**	**RQUGE**	**QA score**
T5-tiny	squad	0.06	0.26	0.08	1.67	-1.34
T5-small	squad	0.19	0.47	0.25	4.16	-1.33
T5-base	squad	0.14	0.40	0.19	3.5	-1.33
T5-large	squad	**0.24**	**0.53**	**0.31**	**4.6**	-1.32
chatGPT-3.5	squad	0.1	0.39	0.19	4.1	**-1.15**
mt5-base	squad_it	0.16	0.38	0.20	-	-
Maximum	squad	1.0	1.0	1.0	4.63	-1.33

Table 1: Scores of Question Generation finetuned models

4.2 Answer Generation

T5-base and T5-small models exhibit comparable performance in answer generation, with T5-base showing a slight edge across BLEU, ROUGE1,

and ROUGE2. However, both T5 models yield relatively modest scores, suggesting that challenges persist in achieving optimal results for answer generation tasks.

Model	**Score**		
	BLEU	**ROUGE1**	**ROUGE2**
T5-small	0.14	0.33	0.22
T5-base	**0.15**	**0.35**	**0.24**
Maximum	1	1	1

Table 2: Scores of Answer Generation finetuned models

4.3 End-to-End Evaluation

In our end-to-end evaluation, we delve into the collective performance of question and answer generation. T5-small and T5-base demonstrate comparable results across BLEU, ROUGE1, and ROUGE2, both for questions and answers.

The pipeline approach, which fuses T5 models for question and answer generation, emerges as a noteworthy strategy. It surpasses individual models in the QA score. However, it's crucial to emphasize a critical aspect of the pipeline approach – the generated answer is employed to subsequently generate the question. This introduces a cascading effect where errors in the generated answer may propagate into the question-generation process.

The evaluation table (see Table 3) provides a detailed breakdown of scores, offering insights into the strengths and weaknesses of each model in the end-to-end context.

Score	Type	T5-small	T5-base	chatGPT	Pipeline	Maximum
BLEU	Question	0.20	0.22	**0.27**	-	1
	Answer	0.09	**0.10**	0.01	-	1
ROUGE1	Question	0.47	0.48	**0.51**	-	1
	Answer	0.36	**0.40**	0.07	-	1
ROUGE2	Question	0.27	0.28	**0.30**	-	1
	Answer	0.23	**0.25**	0.04	-	1
QREL	Question+Answer	**0.16**	0.15	0.09	0.08	0.05
QA score	Question+Answer	-6.63	-5.0	-5.37	**-3.04**	-1.33
RQUGE	Question+Answer	3.63	**4.08**	3.83	3.60	4.63

Table 3: Scores of Answer Generation finetuned models

5 Conclusion

As we conclude our exploration into the realm of question generation using T5 models, we find ourselves enveloped in a tapestry of insights and considerations. From the refined strengths of T5-large to the burgeoning potential of T5-base and T5-small, our journey has encompassed critical introspection on the dimensions of response length, diversity, and the dynamic landscape of evaluation metrics. Moving beyond the confines of model capabilities, we delve into the environmental implications of our computational endeavors, igniting a discourse on the imperative of sustainability in the ever-evolving nexus of technology and research.

5.1 Critical Aspects and Limitations

The investigation into question generation using T5 models and subsequent metric evaluation (RQUGE, QRel, QAScore) reveals strengths and weaknesses. Models tend to generate lengthy answers, requiring a maximum length constraint, raising concerns about impact on content quality. In addition, if an answer-agnostic model is asked to generate multiple question answer pair from the same context, many of them would be repeated with slightly variation. To address this problem strategies to enhance diversity and penalize similar questions should be used, which is crucial

for practical application of these models. Evaluation metrics, like RQUGE and QAScore, lack consideration for grammar correctness, potentially limiting assessment comprehensiveness. These metrics were not designed for answer-agnostic questions, suggesting consideration of alternative evaluation approaches. The application of QRel and QAScore metrics is not clear, therefore we had to rely on the scores obtained from applying them to the original dataset. In conclusion, addressing these critical aspects provides valuable insights, guiding future model improvements and enhancing understanding of generated outputs.

5.2 Carbon footprint

For the experiments we trained in total 44 models, with an average training time of 4 hours each. Our computing infrastructure is based on the NVIDIA Tesla P100 GPUs and is located in Italy, with an average carbon efficiency (kg/kWh) of 0.778 [1]. This is eqivalent of 17.04 kg of CO2 emitted or 69.2 km driven on an average car or 4 beef stakes (of 300g each) or the production of 17.2 litres of milk [2].

5.3 Final consideration

T5-large emerges as a valid approach in question generation, both answer-aware and answer-agnostic, underscoring the importance of model size. T5-base and T5-small demonstrate competitive performance in answer generation, though further refinement may enhance their effectiveness. The pipeline approach showcases potential by combining specialized models for improved end-to-end performance. Evaluation metrics such as QA score, and RQUGE offer a comprehensive perspective, highlighting areas for enhancement in both question and answer generation.

[1] https://www.nowtricity.com/country/italy/

[2] https://www.openco2.net/en/co2-converter

References

[1] Peter F. Brown et al. "Class-Based n-gram Models of Natural Language". In: *Computational Linguistics* 18.4 (1992), pp. 467–480. URL: https://aclanthology.org/J92-4003.

[2] Anthony Chen et al. "MOCHA: A Dataset for Training and Evaluating Generative Reading Comprehension Metrics". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, 2020. DOI: 10.18653/v1/2020.emnlp-main.528. URL: http://dx.doi.org/10.18653/v1/2020.emnlp-main.528.

[3] Jacob Devlin et al. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. 2019. arXiv: 1810.04805 [cs.CL].

[4] Xinya Du, Junru Shao, and Claire Cardie. "Learning to ask: Neural question generation for reading comprehension". In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (2017), pp. 1342–1352.

[5] Liam Dugan et al. "A Feasibility Study of Answer-Agnostic Question Generation for Education". In: *Findings of the Association for Computational Linguistics: ACL 2022*. Ed. by Smaranda Muresan, Preslav Nakov, and Aline Villavicencio. Dublin, Ireland: Association for Computational Linguistics, May 2022, pp. 1919–1926. DOI: 10.18653/v1/2022.findings-acl.151. URL: https://aclanthology.org/2022.findings-acl.151.

[6] Michael Heilman and Noah A Smith. "Good question! Statistical ranking for question generation". In: *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics* (2010), pp. 609–617.

[7] Tianbo Ji et al. "QAScore—An Unsupervised Unreferenced Metric for the Question Generation Evaluation". In: *Entropy* 24.11 (Oct. 2022), p. 1514. ISSN: 1099-4300. DOI: 10.3390/e24111514. URL: http://dx.doi.org/10.3390/e24111514.

[8] Daniel Khashabi, Yeganeh Kordi, and Hannaneh Hajishirzi. *UnifiedQA-v2: Stronger Generalization via Broader Cross-Format Training*. 2022. arXiv: 2202.12359 [cs.CL].

[9] Ghader Kurdi et al. "A Systematic Review of Automatic Question Generation for Educational Purposes". In: *International Journal of Artificial Intelligence in Education* 30 (2020), pp. 121–204.

[10] Linfeng Li et al. "QGAE: an End-to-end Answer-Agnostic Question Generation Model for Generating Question-Answer Pairs". In: *JUSTC* 53 (2023). ISSN: 0253-2778. DOI: 10.52396/JUSTC-2023-0002. URL: https://justc.ustc.edu.cn/en/article/doi/10.52396/JUSTC-2023-0002.

[11] Chin-Yew Lin. "ROUGE: A Package for Automatic Evaluation of Summaries". In: *Text Summarization Branches Out*. Barcelona, Spain: Association for Computational Linguistics, July 2004, pp. 74–81. URL: https://aclanthology.org/W04-1013.

[12] Yinhan Liu et al. *RoBERTa: A Robustly Optimized BERT Pretraining Approach*. 2019. arXiv: 1907.11692 [cs.CL].

[13] Tomas Mikolov et al. "Recurrent neural network based language model". In: vol. 2. Jan. 2010, pp. 1045–1048.

[14] Alireza Mohammadshahi et al. *RQUGE: Reference-Free Metric for Evaluating Question Generation by Answering the Question*. 2023. arXiv: 2211.01482 [cs.CL].

[15] Kishore Papineni et al. "Bleu: a Method for Automatic Evaluation of Machine Translation". In: *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*. Ed. by Pierre Isabelle, Eugene Charniak, and Dekang Lin. Philadelphia, Pennsylvania, USA: Association for Computational Linguistics, July 2002, pp. 311–318. DOI: 10.3115/1073083.1073135. URL: https://aclanthology.org/P02-1040.

[16] Alec Radford et al. "Language Models are Unsupervised Multitask Learners". In: 2019. URL: https://api.semanticscholar.org/CorpusID:160025533.

[17] Colin Raffel et al. *Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer*. 2023. arXiv: 1910.10683 [cs.LG].

[18] Ashish Vaswani et al. "Attention is All you Need". In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon et al. Vol. 30. Curran Associates, Inc., 2017. URL: https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a8
Paper.pdf.

[19] Xiaoqiang Wang et al. *QRelScore: Better Evaluating Generated Questions with Deeper Understanding of Context-aware Relevance*. 2022. arXiv: 2204.13921 [cs.CL].

[20] Sicong Zhao et al. "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* (2018), pp. 3901–3910.

Appendix: Additional Details

Here we report some examples of question generated by ours models

T5 Large - Answer-Aware

Context: Super Bowl 50 was an American football game to determine the champion of the National Football League (NFL) for the 2015 season. The American Football Conference (AFC) champion Denver Broncos defeated the National Football Conference (NFC) champion Carolina Panthers 24–10 to earn their third Super Bowl title. The game was played on February 7, 2016, at Levi's Stadium in the San Francisco Bay Area at Santa Clara, California. As this was the 50th Super Bowl, the league emphasized the "golden anniversary" with various gold-themed initiatives, as well as temporarily suspending the tradition of naming each Super Bowl game with Roman numerals (under which the game would have been known as "Super Bowl L"), so that the logo could prominently feature the Arabic numerals 50.

Reference question: What day was the Super Bowl played on?
Reference answer: February 7, 2016
Generated question: When was the 50th Super Bowl?

Reference question: Which NFL team represented the NFC at Super Bowl 50?
Reference answer: Carolina Panthers
Generated question: Who did the Broncos defeat in the 2015 season?

T5 Base end-to-end - Answer-Agnostic

Context: Six-time Grammy winner and Academy Award nominee Lady Gaga performed the national anthem, while Academy Award winner Marlee Matlin provided American Sign Language (ASL) translation.
Generated question-answer pairs:

- **Q:** Who performed the national anthem?
  A: Lady Gaga
- **Q:** Who sang the national anthem?
  A: Lady Gaga
- **Q:** How many times has Lady Gaga performed the national anthem?
  A: Six
- **Q:** What award did Marlee Matlin win?
  A: Academy Award
- **Q:** How many times has Lady Gaga performed the national anthem?
  A: six

Pipeline: T5 Base Answer - T5 Small Question

Context: In the Philippines, the private sector has been a major provider of educational services, accounting for about 7.5% of primary enrollment, 32% of secondary enrollment, and about 80% of tertiary enrollment. Private schools have proven to be efficient in resource utilization. Per unit costs in private schools are generally lower when compared to public schools. This situation is more evident at the tertiary level. Government regulations have given private education more flexibility and autonomy in recent years, notably by lifting the moratorium on applications for new courses, new schools, and conversions, by liberalizing tuition fee policy for private schools, by replacing values education for third and fourth years with English, mathematics, and natural science at the option of the school, and by issuing the revised Manual of Regulations for Private Schools in August 1992.

Generated questions and answers:

- **Q:** What is the difference between private schools and private schools in the Philippines?
  A: Per unit costs in private schools are generally lower when compared to pub
- **Q:** What percentage of tertiary enrollment does the Philippines have?
  A: 80
- **Q:** What is the percentage of primary enrollment in the Philippines?
  A: about 7.5% of primary enrollment, 32% of secondary enrollment and ab
- **Q:** Are private schools lower or lower than public schools in the Philippines?
  A: per unit costs in private schools are generally lower when compared to pub
- **Q:** What has the private sector been a major provider of education services in the Philippines?
  A: the Philippines, the private sector has been a major provider of education