Preprints.org

Article

Not peer-reviewed version

SynTran-fa: Generating Comprehensive Answers for Farsi QA Pairs via Syntactic Transformation

Farhan Farsi *, Sadra Sabouri, Kian Kashfipour, Soroush Gooran, Hossein Sameti, Ehsaneddin Asgari

Posted Date: 22 October 2024

doi: 10.20944/preprints202410.1684.v1

Keywords: question answering; low-resource NLP; response generation; Farsi NLP

Preprints.org is a free multidiscipline platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This is an open access article distributed under the Creative Commons Attribution License which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 22 October 2024 doi:10.20944/preprints202410.1684.v1

Disclaimer/Publisher's Note: The statements, opinions, and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions, or products referred to in the content.

Article

SynTran-fa: Generating Comprehensive Answers for Farsi QA Pairs via Syntactic Transformation

Farhan Farsi [1,*], Sadra Sabouri [2], Kian Kashfipour [3], Soroush Gooran [4], Hossein Sameti [4] and Ehsaneddin Asgari [5]

[1] Department of Computer Engineering, Amirkabir University of Technology, Tehran, Iran
[2] Thomas Lord Department of Computer Science, University of Southern California, Los Angeles, USA
[3] Department of Computer Science and Engineering, Politecnico di Milano, Milan, Italy
[4] SLPL, Department of Computer Engineering, Sharif University of Technology, Tehran, Iran
[5] Qatar Computing Research Institute - QCRI, Qatar
* Correspondence: farhan1379@aut.ac.ir

Abstract: Generating coherent and comprehensive responses remains a significant challenge Question-Answering (QA) systems when working with short answers especially for low-resourced languages like Farsi. We present a novel approach to expand these answers into complete, fluent responses, addressing the critical issue of limited Farsi resources and models. Our methodology employs a two-stage process: first, we develop a dataset using rule-based techniques on Farsi text, followed by a BERT-based ranking system to ensure fluency and comprehensibility. The resulting model demonstrates strong compatibility with existing QA systems, particularly those based on knowledge graphs. Notably, our system exhibits enhanced performance when integrated with large language models using Chain-of-Thought (CoT) prompting, leveraging detailed explanations rather than single-word answers. Our approach significantly improves response quality and coherence compared to baseline systems. We release our dataset to support further research in Farsi QA.

Keywords: question answering; low-resource NLP; response generation; Farsi NLP

1. Introduction

Recent advances in Natural Language Processing (NLP) resulted in significant progress in various language understanding tasks, with Question Answering (QA) being one of the most crucial ones [1]. QA systems, which aim to generate contextually appropriate and human-like responses to diverse queries, pose unique challenges that intersect multiple aspects of language understanding and generation [2]. Despite substantial progress in high-resource languages [3,4], developing robust QA systems remains particularly challenging for languages with limited computational resources [5].

QA systems, particularly conversational agents, have demonstrated significant impact across diverse domains, including psychological counseling [6], business intelligence [7], and healthcare [8]. The development of large-scale datasets [9–11] has driven substantial progress in this field. Current QA systems tends to output shorter answers which makes it less human, e.g, "*Tehran*" instead of "*The capital of Iran is Tehran*." Large Languge Models (LLMs) tried to solve this issue by adding a Reinforcement Learning with Human Feedback (RLHF) [12] which enforce model to generate longer responses. However, RLHF poses some issues [13,14] which made us to rethink the problem.

Additionally, there are QA systems that only give an entity as output such as question-answering models rooted in knowledge graphs [15], and information retrieval-based models [16]. We propose a novel architecture that augments brief answers into comprehensive responses by jointly conditioning on both the original question and its concise answer. This approach enables QA systems to generate more natural, contextually rich responses while maintaining semantic accuracy. Once these models have generated a short answer, employing our approach simplifies the process of crafting a complete answer. In our research, we have developed a model designed to produce complete and fluent answers from a short answer and corresponding question.

With the development of LLMs, the manner in which we communicate with them has become increasingly crucial. Research indicates that LLMs can enhance the development of reasoning abilities

© 2024 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 22 October 2024 doi:10.20944/preprints202410.1684.v1

when prompted in specific ways. Providing a guided chain of thought [17], rather than solely seeking short answers, enables these models to exhibit advanced reasoning capabilities. Our model can be utilized for prompting to generate guidelines from questions and answers, enhancing the reasoning ability of large language models (LLMs) by producing comprehensive answers rather than brief ones.

2. Dataset

Our goal was to generate a comprehensive dataset that allows researchers to develop models capable of generating complete and fluent responses. Since existing datasets in Farsi were limited to short questions and answers, we applied rule-based approaches to create our dataset. We applied these rule-based methods to a dataset containing pairs of $(question, answer)$, which we discuss in detail later on.

We used datasets containing question-answer pairs, and we focused on incomplete answers. To do this, we employed the Stanza tool [18] to find answers without verbs. We also separated answers with fewer than four words to ensure dataset quality and relevance.

Persian QA dataset [19]. This dataset is the most comprehensive collection in Farsi, covering a wide range of topics. On average, questions contain 6.1 words, showing diversity and complexity. Answers average 8.2 words, indicating detailed and informative responses. In total, the dataset includes 1987 carefully selected question-answer pairs, ensuring its reliability for training our model.

ParSQuAD [20]. In addition to the benefits mentioned earlier, the Persian Short Question-Answer-Dataset (SQUD) contains a large number of concise answers. This dataset includes $63,949$ question-and-answer pairs, making it a valuable and diverse resource for our project. The questions in this dataset have an average word count of 10.73, showing their depth and complexity, while the answers are brief, with an average word count of 2.43 After careful compilation, the final dataset consists of $46,174$ high-quality question-answer pairs, ensuring the reliability of our model training

PQAD dataset [21]. This dataset has proven to be exceptionally valuable for our project, primarily due to its abundant collection of questions accompanied by concise answers. The average word count for questions is 9.8, reflecting the depth and complexity of the inquiries. Conversely, the average word count for answers is merely 2.41, indicating their brevity and straightforwardness. With a substantial compilation of $5,358$ meticulously curated $(question, answer)$ pairs, this dataset offers a diverse and robust resource for training our model.

2.1. Model Workflow for Dataset Generation

The workflow consists of two stages. In the first stage, the question text is modified, and words necessary for a complete question are extracted. In the second stage, the short answer is combined with these words to create a long and fluent response.

2.2. Model Structure

The model structure is divided into two parts. The first step is performed in the first part using rule-based methods based on Persian literature. To generate fluent answers, in the second step a set of candidates is initially generated, and then in the last step, the best answer is selected from them. The following sections explain these stages (Figure 1).

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 22 October 2024 doi:10.20944/preprints202410.1684.v1

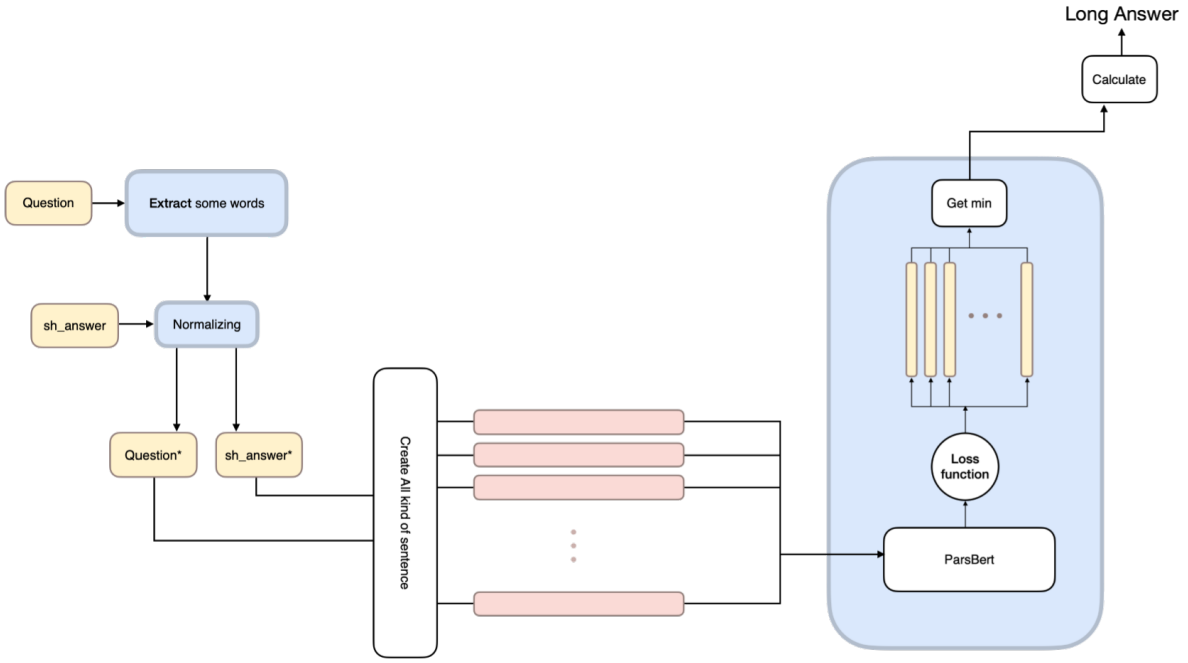

Figure 1. This figure outlines the three-step architecture of our dataset preparation model: (1) word extraction from questions, (2) candidate list generation, and (3) best candidate selection based on ranks.

Word extraction from the question: In order to extract words, we employ a collection of rule-based techniques. We will delve into a comprehensive discussion of these methods in the following.

- Removing extra particles from the words in the question. In the Persian language, the letter ی is used in certain cases. It could be attached to the end of a word and serves various purposes. In questions, the word چه, which means “what”, is co-located with a word that has that letter attached to the end of it. However, in the statement, this particle should be removed for better readability and clarity.
- Removing question words. Naturally, to create a complete answer from the question text, the question words need to be removed from the question text. These words include
  - چند
  - چندم
  - چندمین
  - کجا
  - کی
  - چگونه
  - چرا
  - آیا
  - چی
  - کدام
  - چه
  - چطور
- Converting words from plural to singular. In some cases, the Persian language uses singular forms of words in the answers, contrary to the structure of the questions. This situation occurs in questions that contain the phrase "Which one." The following method is used to convert words from plural to singular: If the word after the phrase is in plural form, it is converted to singular form.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 22 October 2024 doi:10.20944/preprints202410.1684.v1

Table 1. Processed transformation of question text by removing redundant parts.

Question	Question (after processing)
چه تیمی قهرمان شد؟	چه تیم قهرمان شد؟
چه کشوری بزرگترین کشور است؟	چه کشور بزرگترین کشور است؟
چه غذایی در یخچال است؟	چه غذا در یخچال است؟

Creating initial answers: To create possible answers, we utilized a characteristic of the Persian language: the word order in the question and the answer remain the same.

Based on this feature of the Persian language, the best answer generated is undoubtedly the complete and fluent answer. Therefore, the necessary data for the next step is prepared in this stage, and it is required to select the best answer among these answers in the next step. In the Figure 2 the

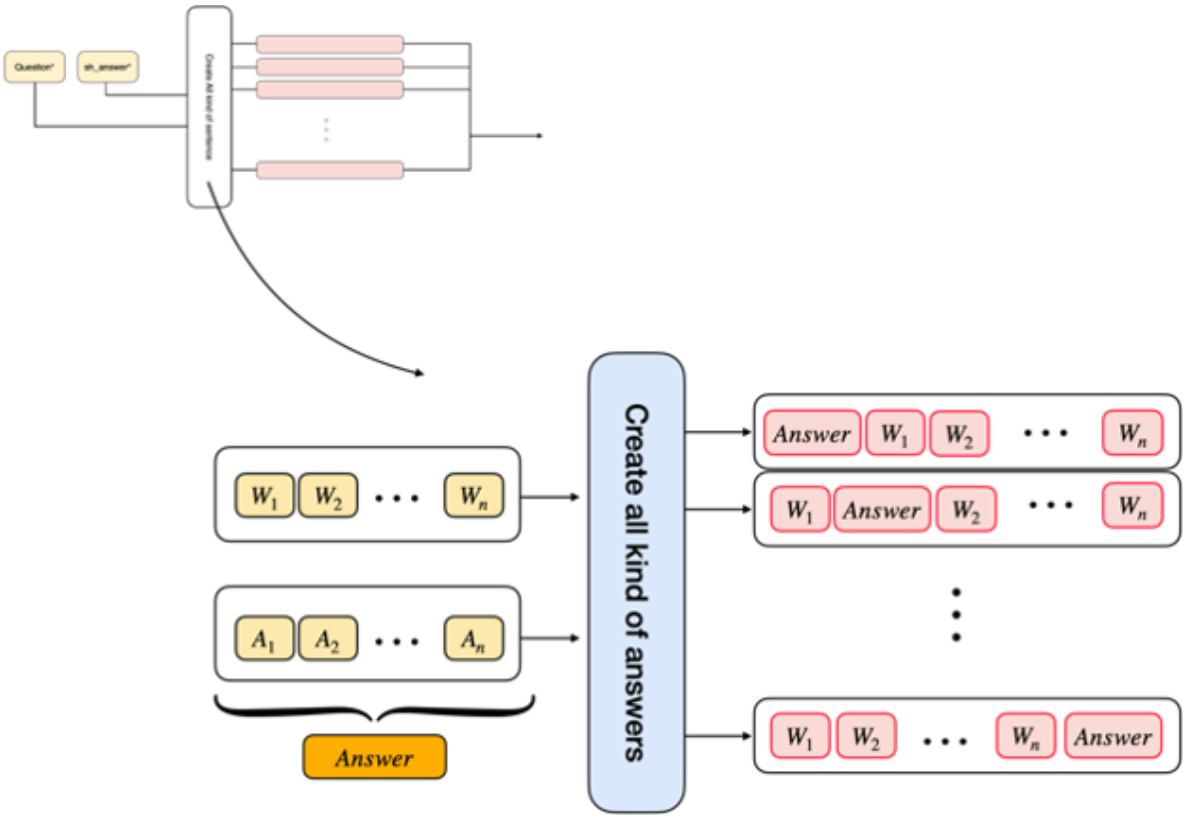

Figure 2. detailed architecture of the second part

Choosing the best answer: In the final stage, the primary objective is to identify the best answer among the candidates generated in the previous stages. Our approach focuses on selecting the answer that is most likely to be correct. To accomplish this, we employed a straightforward method: assigning a loss value to each candidate. Subsequently, we chose the sentence with the lowest loss value as the final answer.

To implement the loss function, we created tensors based on the encoding of each sentence. Then, we generated multiple copies of these tensors, each containing two masked tokens. The model then calculates the log-likelihood of predicting these masked tokens. The ParsBERT model [22] was used as a pre-trained base model in this stage.

2.3. Statistics

In this section, we will take a statistical look at the dataset we have generated. The first important aspect is the diversity of questions, as evident in the Figure 3

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 22 October 2024 doi:10.20944/preprints202410.1684.v1

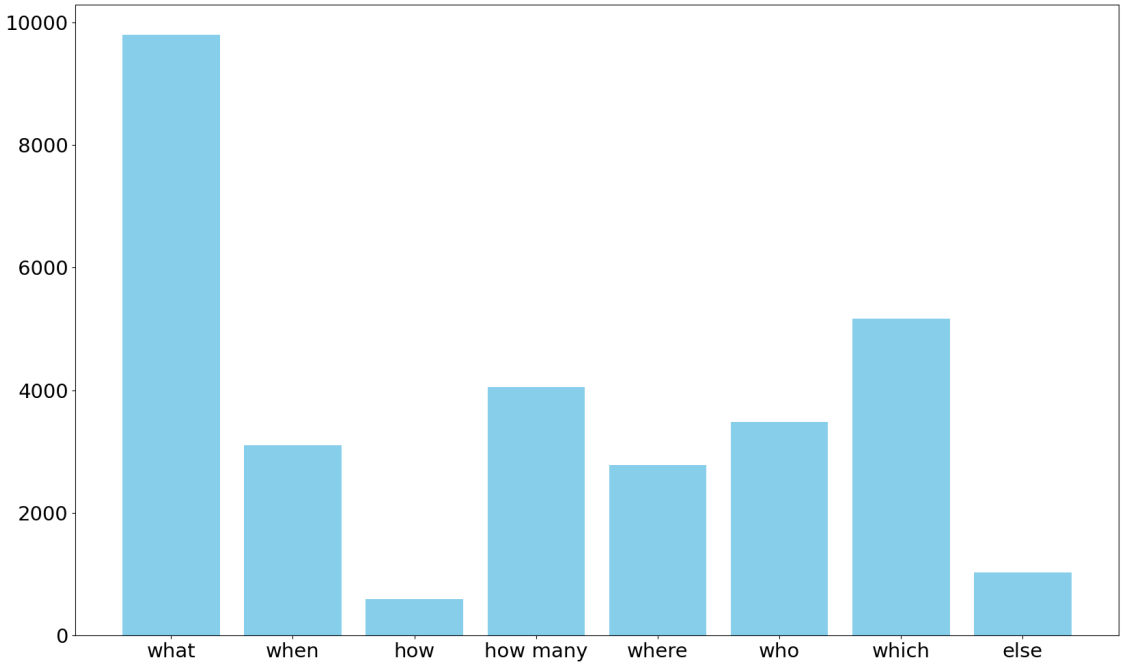

Figure 3. Variety of questions

Considering that the interrogative word "what" has a broader usage, its higher frequency is logical. The fewer questions in the "how" category are because answers to such questions are naturally longer, making it hard to provide short responses. The term "else" includes yes-no questions and others, which are not part of the current project's focus.

Since this project aims to create comprehensive and fluent responses, an appropriate criterion for the success of the model is the length of the generated answers. The statistical distribution of the lengths of the answers is presented in Figures 4 and 5.

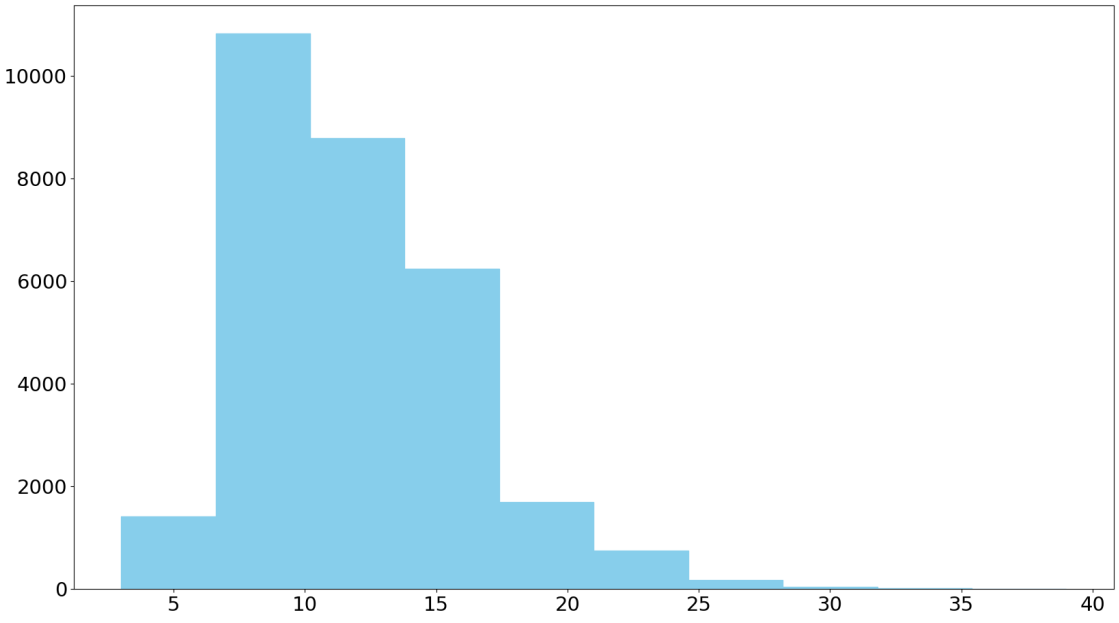

Figure 4. Statistical distribution of the length of complete answers (output)

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 22 October 2024 doi:10.20944/preprints202410.1684.v1

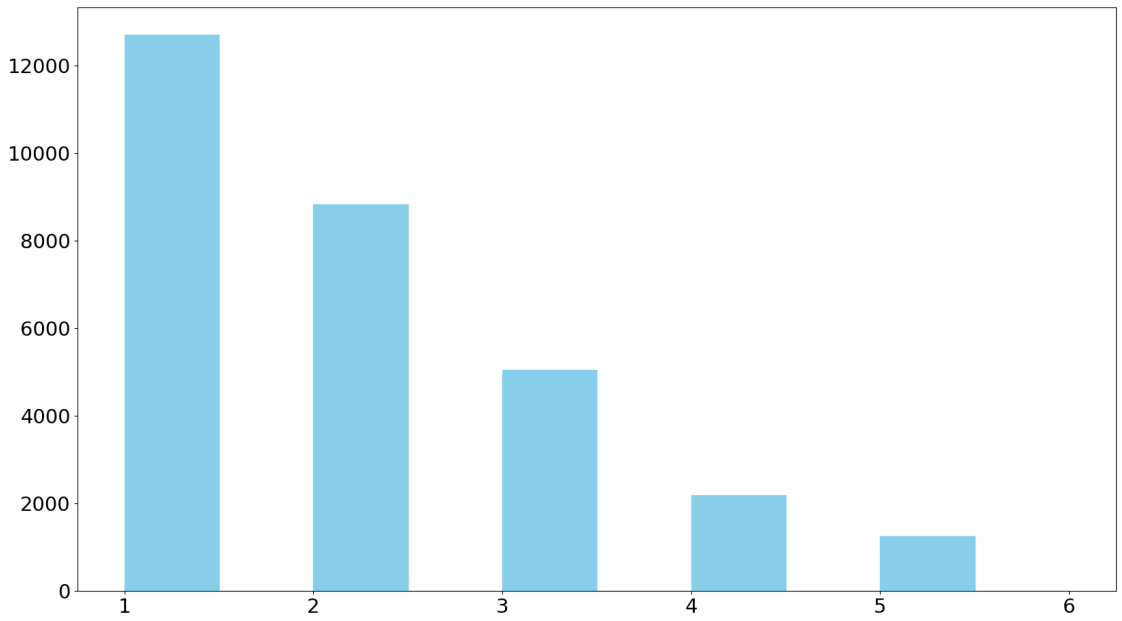

Figure 5. Statistical distribution of the length of short answers

Based on the above chart, the average length of short questions is 2.02, which has been rounded up to 11.93.

3. Model

3.1. Why to Create Model?

At first glance, one might think that, given the dataset creation method discussed earlier, we can use the same approach for new data, and there would be no need to create a new model. However, our previous approach had some unresolved issues, which could only be addressed by developing a new model, creating new data, or making necessary corrections to some data, as explained below. These considerations led us to explore alternative strategies for improved performance.

Our previous model was designed in a way that required generating the necessary words for creating complete answers using some rule-based methods. In the next step, we had to determine how these words should be arranged, which was done using BERT scores. Integrating BERT scores was crucial in determining the optimal word arrangement for more coherent and meaningful answers.

In general, the issues were in two main areas. First, the selection of words had its challenges, which are thoroughly explained below. Second, the BERT score criterion had two problems. First, it was unfamiliar with some words, especially non-Persian words. Second, since it placed greater importance on sentence structure and less on the meanings of words, it sometimes did not select the best answers, which led to suboptimal results. Addressing these concerns is critical for further improving the model's performance. let's delve into the details of the problems with word selection:

Omission of ی in multi-part answers. As mentioned in the dataset creation section, usually, when a word with "Yea" is followed by another word, the "Yea" is omitted in questions. Resolving this problem in multi-part answers was not straightforward because it directly connected with the meanings of words used in the short answers and the questions themselves. Thus, we manually corrected these cases in the first 4000 data points. Additionally, since the model focuses more on sentence structure rather than word meanings and the task is primarily based on grammatical structures rather than word meanings, achieving higher accuracy in selecting the best answers in multi-part short answers

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 22 October 2024 doi:10.20944/preprints202410.1684.v1

proved to be difficult. Despite the challenges faced, the effort to manually correct the dataset's initial portion paid off, resulting in improved performance for the multi-part short answers.

However, it is essential to mention that the trade-off between word meanings and grammatical structures remains a delicate balance to maintain. The model's accuracy in selecting the most suitable answers for multi-part questions still requires further exploration and refinement to achieve even more precise results.

Specific questions. Due to the scarcity of data in some question types, we faced challenges with the following specific questions, as the model did not comprehend them well and did not perform satisfactorily:

- How much of
- How many of
- Which one of
- How many times
- What year

To handle this issue, we generated approximately 50 samples for each question type, and we achieved good results from the model.

converting plural form to the singular form: Another issue we encountered was related to plural words in certain types of questions in the Persian language. Although the answer required the singular form of the word to appear in the sentence, the existence of plural forms in Persian made it difficult to identify such cases using rule-based methods. To resolve this error, we added additional data points to the dataset.

3.2. Model Setup

To emphasize both input and output, we made the decision to employ an encoder-decoder-based transformer architecture. Consequently, we utilized the MT5 [23] model as our chosen model, and the mt5-base-finetuned-persian eslam-xm was employed as our foundational pre-trained model. While conventional recommendations indicate that the fine-tuned model's learning rate should typically be one-thousandth of the pre-trained model's learning rate, we achieved better results using the same learning rate as the pre-trained model. so we assigned the learning rate of 5.10^{-4} and trained the model for 5 epochs. we used Adamw optimizer [24] furthermore the outputs were generated by beam search method [25] with a value of 5 for num-beams and a value of 3 for the parameter top-k and 0.90 for the top-p parameter

3.3. How to Integrate Inputs

One of the most challenging parts of this research was how to integrate inputs. T5 accepts only one input, and our task requires two inputs, we needed to find a way to connect these inputs. Initially, we tried the approach of:

question + [SEP] + short answer

which provided reasonably good outputs. However, a problem arose when dealing with multi-part answers. In such cases, the long answer, which essentially represents the complete response, did not include the short answer, rendering the long answer useless. The reason for this problem was that when padding tokens followed the short answer, the model focused more on the padding rather than the actual answer, leading to inaccurate outputs.
To resolve this issue, we modified the input by using three [SEP] separators in the following manner:

[SEP] + question + [SEP] + short answer + [SEP]

With this approach, we could solve the problem and have the long answers include the short answers in the outputs. The three [SEP] separators enabled the model to distinguish and utilize

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 22 October 2024 doi:10.20944/preprints202410.1684.v1

both question and short answer information effectively, improving the overall performance of our model in generating meaningful and comprehensive responses for multi-part questions. Nonetheless, ongoing research and experimentation are required to fine-tune the model further and explore potential variations in input formatting to optimize results.

The method mentioned above provided satisfactory results, but another potential problem was truncation. To address this, we adopted the approach of swapping the positions of the question and short answer. In the event of truncation, the short answer, which is essentially the central part of the long answer, was not removed, allowing us to retain the less important parts of the question at the end. So, the method we've finalized is presented below:

[SEP] + short answer + [SEP] + question + [SEP]

The loss values are provided for various methods in Table 2.

Table 2. Model loss after five epochs for different forms of question and answer integration.

Integration Method	Loss (after 5 epochs)
$sh\_answer + [SEP] + question$	0.127
$[SEP] + question + [SEP] + sh\_answer + [SEP]$	0.103
$[SEP] + sh\_answer + [SEP] + question + [SEP]$	0.109

As evident, there is no significant numerical difference between different methods, as the variations are generally limited to 1 or 2 words, usually corresponding to the short answers. However, these words are crucial as an extended answer without these short phrases would be practically meaningless. The evaluation metric, however, does not consider the importance of these short phrases, making the integration of results highly significant despite the close loss values.

4. Results

We evaluated the performance of our model using three metrics: METEOR, BLEU, and ROUGE. Precision was a key consideration, especially in ROUGE, because our primary goal was to ensure that all information, particularly the words from the questions, was accurately and comprehensively included in the generated answers. It was important that the essential content of the questions was covered, even if some additional words appeared in the answers. The results are shown in Table 3.

Table 3. Performance metrics of the model.

BLEU				METEOR	ROUGE		
BLEU-1	BLEU-2	BLEU-3	BLEU-4		ROUGE-1	ROUGE-2	ROUGE-L
0.823	0.745	0.672	0.612	0.911	0.850	0.800	0.872

5. Applications

In this section, we delved into the developmental contributions our model extends to other modules. Now, our focus shifts to chatbots, the quality of QA datasets, and prompting mechanisms.

QA systems: One of our applications involves integrating our model with question-answering systems. Our model seamlessly aligns with task-based chatbots, where the typical objective is to provide clients with specific short answers. By incorporating our model into this context, responses can adopt a more natural and human-like tone. Additionally, for question-answering systems that exclusively output entities rather than complete sentences, our model excels in generating coherent sentences, enhancing the overall conversational quality.

Prompt engineering: In the field of language processing, large language models are quite important. How we deal with them matters, and our model stands out because it can create better cues. It's good at this because it provides more helpful information to the large language model, making the communication and results more detailed and insightful. Here we did zero-shot learning in relation

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 22 October 2024 doi:10.20944/preprints202410.1684.v1

extraction task on GPT-3.5 Turbo. In this approach, we provide the model with more information only from the question, making it wiser and enhancing its ability to understand things. In Figure 6a we can see that ChatGPT can easily answer the question using the prompt that was made by our model. In Figure 6b ChatGPT was unable to answer the question without the help of our model.

LLMs datasets: The field of Large Language Models (LLMs) has seen significant advancements, with datasets playing a crucial role in their performance. In particular, SynTran-fa datasets are currently used to train various LLMs such as [26]. Since our data is prepared from question-answering datasets based on Wikipedia data, it can improve the knowledge of the LLM in the Persian language on which we have a lack of datasets.

(a) Answer of Chat GPT to the prompt our model creates

(b) Answer of Chat GPT to the usual prompt

Figure 6. Comparison of responses from Chat GPT

6. Suggestions

The created model has a good performance, but there are areas where it can be improved. One such area is the model's inability to utilize punctuation marks. Incorporating punctuation marks would significantly enhance the fluency of the generated answers, an aspect not currently handled well by our model. Additionally, our model is currently limited to answering yes-or-no questions, and more complex queries, such as "this" or "that," are not within its capabilities. This limitation can be addressed by creating a dataset that includes such questions.

Furthermore, our model generates the most comprehensive answer possible, attempting to utilize all the information given in the question. However, including all this information in the answer is not

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 22 October 2024 doi:10.20944/preprints202410.1684.v1

always necessary. Making the model more selective in choosing which information to include can be an area of improvement.

7. Conclusions

Our work makes several key contributions to question-answering systems by decomposing the response generation process into two complementary phases: answer extraction and response synthesis. This decomposition not only enables more precise control over answer generation but also significantly enhances the capabilities of existing knowledge graph-based and entity retrieval systems, allowing them to produce fluent, contextually appropriate responses. Our approach demonstrates particular efficacy in production environments, where it can be seamlessly integrated with existing chatbot architectures to improve response quality. Moreover, the proposed framework substantially reduces the annotation burden in dataset creation by requiring only concise answers, while our paraphrasing mechanism automatically generates diverse, contextually rich responses. This work opens new avenues for developing more sophisticated QA systems, particularly for low-resource languages and specialized domains where comprehensive training data may be limited.

References

1. Mervin, R. An overview of question answering system. *International Journal Of Research In Advance Technology In Engineering (IJRATE)* **2013**, *1*, 11–14.
2. Kodra, L.; Meçe, E.K. Question answering systems: A review on present developments, challenges and trends. *International Journal of Advanced Computer Science and Applications* **2017**, *8*, 217–224.
3. Kwiatkowski, T.; Palomaki, J.; Redfield, O.; Collins, M.; Parikh, A.; Alberti, C.; Epstein, D.; Polosukhin, I.; Devlin, J.; Lee, K.; others. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics* **2019**, *7*, 453–466.
4. Qiao, T.; Dong, J.; Xu, D. Exploring human-like attention supervision in visual question answering. Proceedings of the AAAI conference on artificial intelligence, 2018, Vol. 32.
5. Magueresse, A.; Carles, V.; Heetderks, E. Low-resource languages: A review of past work and future challenges. *arXiv preprint arXiv:2006.07264* **2020**.
6. Yin, J.; Chen, Z.; Zhou, K.; Yu, C. A deep learning based chatbot for campus psychological therapy. *arXiv preprint arXiv:1910.06707* **2019**.
7. Waghmare, C.; Waghmare, C. Deploy chatbots in your business. *Introducing Azure Bot Service: Building Bots for Business* **2019**, pp. 31–60.
8. Dharwadkar, R.; Deshpande, N.A. A medical chatbot. *International Journal of Computer Trends and Technology (IJCTT)* **2018**, *60*, 41–45.
9. Rajpurkar, P.; Zhang, J.; Lopyrev, K.; Liang, P. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250* **2016**.
10. Clark, C.; Lee, K.; Chang, M.W.; Kwiatkowski, T.; Collins, M.; Toutanova, K. BoolQ: Exploring the surprising difficulty of natural yes/no questions. *arXiv preprint arXiv:1905.10044* **2019**.
11. Joshi, M.; Choi, E.; Weld, D.S.; Zettlemoyer, L. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. *arXiv preprint arXiv:1705.03551* **2017**.
12. Ouyang, L.; Wu, J.; Jiang, X.; Almeida, D.; Wainwright, C.; Mishkin, P.; Zhang, C.; Agarwal, S.; Slama, K.; Ray, A.; others. Training language models to follow instructions with human feedback. *Advances in neural information processing systems* **2022**, *35*, 27730–27744.
13. Kutalev, A.; Markoff, S. Investigating on RLHF methodology. *arXiv preprint arXiv:2410.01789* **2024**.
14. Sheng, G.; Zhang, C.; Ye, Z.; Wu, X.; Zhang, W.; Zhang, R.; Peng, Y.; Lin, H.; Wu, C. HybridFlow: A Flexible and Efficient RLHF Framework. *arXiv preprint arXiv:2409.19256* **2024**.
15. Yasunaga, M.; Ren, H.; Bosselut, A.; Liang, P.; Leskovec, J. QA-GNN: Reasoning with language models and knowledge graphs for question answering. *arXiv preprint arXiv:2104.06378* **2021**.
16. Yadav, V.; Sharp, R.; Surdeanu, M. Sanity check: A strong alignment and information retrieval baseline for question answering. The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, 2018, pp. 1217–1220.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 22 October 2024 doi:10.20944/preprints202410.1684.v1

17. Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; Xia, F.; Chi, E.; Le, Q.V.; Zhou, D.; others. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems* **2022**, *35*, 24824–24837.
18. Qi, P.; Zhang, Y.; Zhang, Y.; Bolton, J.; Manning, C.D. Stanza: A Python natural language processing toolkit for many human languages. *arXiv preprint arXiv:2003.07082* **2020**.
19. Ayoubi, Sajjad & Davoodeh, M.Y. PersianQA: a dataset for Persian Question Answering. https://github.com/SajjjadAyobi/PersianQA, 2021.
20. Abadani, N.; Mozafari, J.; Fatemi, A.; Nematbakhsh, M.; Kazemi, A. Parsquad: Persian question answering dataset based on machine translation of squad 2.0. *International Journal of Web Research* **2021**, *4*, 34–46.
21. Darvishi, K.; Shahbodaghkhan, N.; Abbasiantaeb, Z.; Momtazi, S. PQuAD: A Persian question answering dataset. *Computer Speech & Language* **2023**, *80*, 101486.
22. Farahani, M.; Gharachorloo, M.; Farahani, M.; Manthouri, M. Parsbert: Transformer-based model for Persian language understanding. *Neural Processing Letters* **2021**. doi:10.1007/s11063-021-10528-4.
23. Xue, L.; Constant, N.; Roberts, A.; Kale, M.; Al-Rfou, R.; Siddhant, A.; Barua, A.; Raffel, C. mT5: A massively multilingual pre-trained text-to-text transformer. *arXiv preprint arXiv:2010.11934* **2020**.
24. Loshchilov, I.; Hutter, F. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101* **2017**.
25. Graves, A. Sequence transduction with recurrent neural networks. *arXiv preprint arXiv:1211.3711* **2012**.
26. Abbasi, M.A.; Ghafouri, A.; Firouzmandi, M.; Naderi, H.; Bidgoli, B.M. PersianLLaMA: Towards Building First Persian Large Language Model. *arXiv preprint arXiv:2312.15713* **2023**.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.