ISSN 2007-9737

Transformer-Based Extractive Social Media Question Answering on TweetQA

Sabur Butt[1], Noman Ashraf[1], Muhammad Hammad Fahim Siddiqui[2], Grigori Sidorov[1], Alexander Gelbukh[1]

[1] Instituto Politécnico Nacional,
Centro de Investigación en Computación,
Mexico

[2] University of Ottawa,
School of Electrical Engineering and Computer Science,
Canada

{saburbutt, hammadfahim, nomanashraf}@sagitario.cic.ipn.mx,
sidorov@cic.ipn.mx, gelbukh@gelbukh.com

Abstract. The paper tackles the problem of question answering on social media data through an extractive approach. The task of question answering consists in obtaining an answer from the context given the context and a question. Our approach uses transformer models, which were fine-tuned on SQuAD. Usually, SQuAD is used for extractive question answering for comparing the results with human judgments in social media TweetQA dataset. Our experiments on multiple transformer models indicate the importance of application of pre-processing in the question answering on social media data and elucidates that extractive question answering fine-tuning even on other type of data can significantly improve the results reducing the gap with human evaluation. We use ROUGE, METEOR, and BLEU metrics.

Keywords. Question answering, SQuAD, tweetQA, social media, tweets.

1 Introduction

Social media is a major source of multiple scientific studies on natural language processing (NLP) problems since it gives deep insight into the way people speak, think and react to multiple social settings. On several datsets, question answering (QA) systems have surpassed human evaluations in extractive settings.

Hence, to make question answering datasets more challenging, several researchers have tried to build more open-ended, high context, and multiple hop reasoning datasets [9, 13, 24]. Most of these question answering systems are made from formal documents like Wikipedia, fiction stories, and news. Therefore, there is a dearth of question answering systems that do well with informal social reading comprehensions.

A recently proposed question answering dataset TweetQA [22] collected questions and answers over social media Twitter data. Designing a question answering system from informal social media setting has left us with a unique problem of inferring answers from multiple short sentences. Since tweets are only limited to 280 characters, we have few context and there are many tweet specific (informal) expressions.

The differentiating factors of tweets in the question answering problem are that firstly, it involves a large number of user accounts with heavy tails that requires an understanding of how tweets are related to the authors. Secondly, tweets are mostly informal and require the understanding of common oral language. Lastly, it requires an understanding of tagged IDs and hashtags, which are single special tokens. Understanding these helps answer event related or person related questions. We also have baselines for TweetQA results. These are commonly used neural baselines, which give excellent results on existing formal text datasets.

doi: 10.13053/CyS-25-1-3897

There are various types of models for QA task. If we consider the SQuAD dataset (Stanford Question Answering Dataset), we can meniton teh following approaches. The generative model proposed by [18] gave the BLEU-1 score of 53.7, METEOR score of 31.8, and ROUGE-L score of 38.6. Bidirectional attention flow [17] gave the BLEU-1 score of 48.7, METEOR score of 31.4, and ROUGE-L score of 38.6. Finally, the fine-tuned BERT [4], which has the best score on SQuAD, gave the BLUE-1 score of 61.4, METEOR score of 58.6, and ROUGE-L score of 64.1. All these scores are relatively low comparing to the achieved scores on other available datasets. In this paper, we discussed the impact of pre-processing on the informal social media tweets.

Further in the paper, we discuss the challenges posed by social media question answering systems and how we can deal with them to find better results. We used transfer learning methods trained on SQuAD 2.0 dataset to find out how the trained algorithms that work very well in extractive settings on formal data would perform in a social media context. We were able to show that fine-tuned algorithms like ALBERT [10] work very well getting closer to the human evaluation.

The paper has the following structure. In the section two, we discuss the relevant work done on question answering and the methods used to obtain human competitive results. The third and fourth sections explain the methodology and the results, while the fifth section concludes the findings of the paper.

2 State of the Art

We studied the state of the art models for SQuAD 2.0, which is a standard benchmark for extractive question answering tasks. We found out that most of the models topping the charts were essentially the variants of ALBERT [10] or complicated attention based document readers. For models to work better than the human evaluation on SQuAD, they need to be designed to counter situations, when no answer is expected from the given context. This trend shifted from attention based interactions between question and passage to pre-trained language models with robust encoder settings.

Gated self matching [21] passes the context and question through the bidirectional recurrent network (BRNN) and matches them by the gated attention based recurrent network (GRNN), and applies self-matching attention to get the refined answers. Attention sum [8] is another example, in which the attention was designed to directly pick the word as an answer from the context by using a dot product between the contextual and question embeddings of each occurrence. Gated attention [5] brings an attention mechanism, when the intermediate states of recurrent neural network reader and query embeddings have multiplicative interactions to obtain accurate answers using query specific representations.

Bi-attention [17] changes question and context into vectors embeddings and then uses attention to merge information contained in the question and the context to create bidirectional context aware query representation, which enhances the attention span. The state of the art retrospective reader design in SQuAD [26] further combines verification and reading steps by inculcating sketchy reading and intensive reading for initial and final judgment.

ALBERT [10] is popular, because it superseded BERT [4] with fewer parameters and achieved +3.1% more accuracy on SQuAD 2.0 benchmark. ALBERT is designed like BERT transformer encoder structure, however, unlike BERT paper, which argues that more hidden layers, hidden sizes, and attention heads can improve the results, ALBERT paper showed that it was not the case through detailed experimentation. ALBERTS popularity compared to architectures [12, 23] is due to cross layer parameter sharing, sentence order prediction and factorizing embedding parameters.

Another variant of BERT is spanBERT, which gives exceptional results on SQuAD 2.0 and advances BERT by masking adjacent spans instead of tokens and training span boundary for prediction of the entire masked span. This technique works well for abstractive question answering.

2.1 Comparison of Available Datasets

Machine reading comprehensions are expected to answer questions based on the content of the documents or the supporting context, which is

Table 1. Examples of TweetQA obtained from validation set

Context: The #endangeredriver would be a sexy bastard in this channel if it had water. Quick turns. Narrow. (I'm losing it) John D. Sutter (@jdsutter) June 21, 2014
Question: What is this user losing? Answer: "he is losing it", "it"
Context: Photo: In the cell of one of escaped inmates, looking at the hole cut in the wall as part of escape Andrew Cuomo (@NYGovCuomo) June 6, 2015
Question: How did the inmates escape? Answer: "a hole cut in the wall.", "cut in the wall"

Table 2. Examples of pre-processing errors in social media question answering

Question	Correct answer	Answer after pre-processing
How big of an increase does alaska have in obamacare premiums?	200%	200
What time was this video posted?	8:20 am	820 am
How often for trains?	8-10 min	810 min
What historical date is mentioned in the tweet?	9/11	911

provided to determine answers. We can divide these answers into two categories: abstractive and extractive. Extractive question-answer datasets such as SQuAD [15], NewsQA [19], TRECQA [20], CoQA [16], SearchQA [6], and QuAC [2] focus on extracting the span of answers from the context or document and evaluate using F1 or ROUGE with the ground truth span.

We can also find datasets like HOTPOTQA [24] which are designed to conduct multi-hop reasoning across multiple paragraphs, but the answer is still extractive in nature. Whereas, abstractive question answer datasets such as NarrativeQA [9], TweetQA [22], CoQA [16], TriviaQA [13], and QuAC [2] are designed to answer questions that may not appear in the passage. Every dataset is distinct and tries to tackle a variety of question answering problems, i.e., unanswerable questions, multi-turn interactions, abstractive answers, conversational context, and multihop reasoning. Below, we present the summaries of some leading and important extractive and abstractive datasets:

— Stanford Question Answering Dataset(SQuAD 2.0) [15] deals with extractive question answering, whereby the dataset provides context paragraph retrieved from Wikipedia. The paragraph in the dataset either have relevant or plausible answers to the questions. Unanswerable questions are relevant to the topic and answerable questions have a span of text in the paragraph, which is considered as the correct answer. SQuAD 2.0 has 100,000 questions with exact answer spans and over 50,000 unanswerable questions written to look similar to answerable ones. SQuAD 2.0 is challenging as systems must also answer in cases, when no answer is supported by the paragraph or abstain from answering.

— NEWS QA [19] has 100,000 question-answer pairs based on over 10,000 news articles from CNN news with answers, which are extractive in nature. SQuAD superseded the dataset in every way in the second version and the only remained difference was of the source of context, which in the case of NEWS QA was news centric.

— HOTPOTQA [24] is a dataset with 113K question answering pairs based on Wikipedia. The dataset is different as the questions require finding and reasoning over multiple supporting

Table 3. Experiment results on TweetQA with pre-trained models on SQuAD 2.0. (*) shows results trained without # and @ sign in the tweets. See Table 4 for results with direct training

Model Name	BLEU-1	METEOR	ROUGE-L
Human Performance	78.2	66.7	73.5
BERT	66.97	64.60	68.85
BERT *	66.30	63.73	68.23
ALBERT	68.41	65.86	70.56
ALBERT *	**69.03**	**66.35**	**71.28**
SpanBERT	61.64	58.68	64.06
SpanBERT *	60.86	57.70	63.50

Table 4. Experiment results on extractive answer spans trained on TweetQA. (*) shows the leading results of the TweetQA challenge (without description or code available)

Model Name	BLEU-1	METEOR	ROUGE-L
Human Performance	78.2	66.7	73.5
BERT	69.6	58.6	64.1
ALBERT	64.02	61.75	66.42
PingAnLifeInsuranceAI*	73	70	75

documents to answer. The questions are diverse and not constrained to any preexisting knowledge bases, thus, the dataset provides sentence level supporting facts required for reasoning. It provides a factoid comparison questions to test QA systems. The structure of the HotpotQA is such that it gives explainability of question answering systems, by output of a set of supporting facts necessary to arrive at the answer, when the answer is generated. These supporting facts hence serve as strong supervision for sentences to pay attention to.

— Trivia QA [13] is a reading comprehension dataset containing over 600K question answers with context. There are six documents on average per question for distant supervision of answers. TriviaQA is different as it had complex and compositional questions. Secondly, it has considerable syntactic and variability between questions and corresponding answer evidence sentences. It also requires more cross sentence reasoning to find answers. The questions in TriviaQA are authored organically and the context documents are collected retrospectively from the Wikipedia and the web. TriviaQA was designed to engage humans, since the dataset should be able to deal with a large amount of text from various sources such as encyclopedic entries, blog articles, and news articles should handle inference over multiple sentences.

— Conversational Question Answering (CoQA) [16] is a dataset designed in the form of conversations. It contains 127K unanswerable questions with answers, extracted from 8K conversations about certain text passages from seven different domains namely children stories, literature, Wikipedia, Reddit, science, mid/high school exams, and news. CoQA ensures naturalness of answers, system robustness across domains, and questions that depend on conversation history.

— NarrativeQA [9] is a dataset consisting of stories, which are books and movie scripts, with questions and answers written by humans. The questions and answers are based solely on human generated abstractive summaries. For the reading comprehension tasks, questions may be answered using the full story text or just the summaries.

Table 5. Example of SQuAD, when SQuAD understands hashtags better

Context: Started researching this novel in 2009. Now it is almost ready for you to read. Excited! #InTheUnlikelyEvent Judy Blume (@judyblume) December 15, 2014 Answer: “in the unlikely event.”
Question: what is the name of the novel? Answer generated by SQuAD: “#InTheUnlikelyEvent” Answer generated by TweetQA: “2009”

Table 6. SQuAD model sentence relation and deep semantics error

Context: Sorry guys, rant over. I hope you all had a great day. If I wasn’t able to vent to you guys, these situations would probably end in violence Cara Delevingne (@Caradelevingne) September 22, 2015 Answer: “for ranting”
Question: what is this person apologizing for? Answer generated by SQuAD: “Sorry” Answer generated by TweetQA: “rant overs”

— Question Answering in Context (QuAC) [2] is a context oriented dataset that contains 14K information seeking question answering dialogues. The dialogue involves free form questions posed by a Crowdworker about a hidden Wikipedia text. The second Crowdworker answers the question by providing short excerpts from the text. QuAD is different because of the open-ended, unanswerable, and meaningful withing the dialogue context.

In a qualitative comparison of CoQA, SQuAD, and QuAC, it was claimed [25] that there are three distinct features of the dataset, namely, unanswerable questions, multi-turn interactions, and abstractive answers.

It was argued that model trained on one dataset is ineffective on the other, because no dataset provided significant coverage of abstractive answers making it a challenge as compared to the extractive spanning of answers. TweetQA [22], therefore, becomes a very suitable dataset to evaluate extractive pretrained dataset on abstractive answers in a social context.

While SQuAD provided a very balanced dataset of unanswerable and answerable questions, it was interesting to see how models trained on SQuAD would perform on TweetQA environment based on feature similarities and differences.

Table 1 shows examples from TweetQA validation set. TweetQA contains 42.33% “What” and 29.36% “Who” type questions. Apart from the TweetQA leaderboard, no results have been published previously using BERT, ALBERT and other transformer methods on the task of social media question answering.

3 Experiment Details

We conducted our experiments on the TweetQA test set with various leading models pretrained on SQuAD 2.0. Our experiments show results for BERT, ALBERT and SpanBERT. We also show the impact of the hashtag, “at” symbol, and emojis in the Twitter test set.

3.1 Pre-processing of Tweets

It is important to understand the challenges with tweets in question answering, since pre-processing in a question answering task can vary largely than other natural language processing problems. Table 2 shows various examples, when pre-processing techniques like emoji, hashtag and “at” symbol

Table 7. Expected answer is not part of the context as a substring

Context: Words will never be enough to justify the connection we shared and the pain I will forever feel. RIP @Knight_MTV Jemmye Carroll (@JustJem24) November 27, 2014 Answer: “lost a significant other, someone died ”
Question: Why is the tweeter in pain? Answer generated by SQuAD: “2014” Answer generated by TweetQA: “@Knight”

Table 8. Correct understanding of the tweetQA model, but paraphrased gold answer

Context: Just because I haven’t used AIM in years doesn’t mean I am not sad. It is like finding out a band you were really into 15 years ago broke up. Mute Bae (@DanGnajerle) October 6, 2017 Answer: “stuff that has been gone ofr 15 yars, aim going away”
Question: what makes us sad? Answer generated by SQuAD: “2017” Answer generated by TweetQA: “broke up”

removal and Unicode conversion of text can contribute negatively in evaluating question answering results.

Similarly, URLs, emails, phone numbers, general numbers and currency symbols might be irrelevant in tasks like emotion detection [1] but hold great importance in question answering. These attributes of the tweets are potential answers and a single missing punctuation can change the value of the price being asked in a question. Hence for all the experiments, we convert the sentences into either word or sentence pieces depending on the model.

The normalization for the evaluation of text has the following steps:

— Removing extra white spaces from the answers.

— Lowering the text as it facilitates in extracting answers and maintaining consistency in text.

— Removing stopwords (articles, prepositions, etc.) within the text, i.e., *and*, *a*, *the*, etc.

Since the TweetQA dataset consists of paraphrased answers, it is not possible to obtain the Startidx and the Endidx of the answers as a span from the context. We removed the instance in the TweetQA data that was not the exact span from the data. After the omission of the non-span answers from the tweet, we obtained 5,564 instances to train our model.

3.2 Fine-Tuned Models

We used HuggingFace pretrained models for the testing of our results. We evaluated our data using three metrics, namely, BLEU-1 [14], Meteor [3], and Rouge-L [11]. The evaluation is done by comparing the answers generated by the system and gold answers annotated by humans. Below we mention the parameters and fine-tuning details of every model used in the experiment.

3.2.1 BERT

BERT large model[1] was used as described in the BERT paper with dropout probability of 0.1, 16 attention heads, 24 hidden layers and 340M parameters. The input context and questions were converted into Wordpiece embeddings and later packed into a sequence. BERT format of input “[CLS] + tokens + [SEP] + padding” was used as described in the paper. We used HuggingFace BERT for question answering for the implementation and training of our model.

[1] `https:\\gist.github.com\saburbutt`

3.2.2 ALBERT

Albert xxlarge model[1] with $H = 4096$ and 233M parameters was used originally in the ALBERT paper [10] was fine tuned with 12 hidden layers, dropout with $p = 0.1$, 64 attention heads and 512 positional embeddings. We use both question and context as input using sentencePiece tokenization and the text was formatted using ALBERTS standard example using "[CLS] x1 [SEP] x2 [SEP]" format, where x1 and x2 are two segments. Tokenization was done using HuggingFace tokenizer for Albert.

3.2.3 SpanBERT

SpanBert large model[1] was used as described in SpanBERT paper [7] using span boundary objective, span masking and single sequence training. We set attention heads to 12, hidden layers to 12, dropout to 0.1 and positional embeddings to 512. We first converted the passage and question into sequences with the format as defined in the paper, passing the input into transformer encoder and trained two linear classifiers independently on top of it for span prediction of answers.

4 Results

Table 3 shows that ALBERT model solely trained on SQuAD 2.0 in an extractive setting was able to close the gap between human evaluation. While other models illustrated a slight decrease in all evaluation methods with the removal of # and @, Albert results improved as compared to the regular setting obtaining 69.03 BLEU-1, 71.28 ROUGE-L, and 66.35 METEOR score.

Results of all other tested models reduced after the extraction of hashtag and "at sign". Since some answers in the TweetQA dataset contained hashtag and "at sign", the decrease was rather expected. The second best BERT score although achieved 66.30 BLEU-1 score, but still has a lot of vacuum to cover in order to achieve the human performance.

If we analyze the results shown for TweetQA in Table 4, we see that the results for extractive answers dropped significantly on the same models. The first difference is of the size of the dataset that we used for training. We also see the leading result in the challenge, however, none of the methods[2] used for TweetQA are scientifically available, i.e., no method description is given nor the code is available.

SQuAD in comparison of TweetQA has more number of instances to train on, hence, the encoder establishes a better understanding of the trained data and performs better fine-tuning on similar tasks. On the other hand, TweetQA did work good with the abstractive transformer models and hence underperformed due to the abstractive settings of the answers.

4.1 Error Analysis

We conducted a detailed error analysis on all the development set of TweetQA to analyze the instances, in which transformer models completely missed. We used BERT Large for the experiment. The results show the type of mistakes generated by SQuAD and TweetQA trained model.

The error analyses show us that TweetQA's informal structure of sentences and language makes it a little difficult to understand the sentence relations and deep semantics based questions and contexts. For the models trained on SQuAD, they were able to understand successfully social media attributes such as Hashtags and UserIDs, among others. The performance of understanding of informal languages also varied since all models have different training backgrounds.

One of the other major issue seen in SQuAD trained model is in the common sense prediction answers, when the model predicts dates and numbers instead of real relations described in the context. This particular problem improves when the bigger models are used and the parameters are better set to achieve higher accuracy, i.e., Albert xxlarge. Examples of these are seen in Table 6. We can also see instances, when the gold answer is not part of the substring as presented in Table 7.

The ambiguous nature of the questions sometimes is misleading as the model understand the gist of the question, but it is not what is expected out of the answers. A perfect example of that

[2] `https://competitions.codalab.org/competitions/20307#results`

Table 9. Example of TweetQA common sense error

Context: Very saddened by the news Jordan Feldstein passed. He was such a character & will be sorely missed by many. May he Rest In Peace. IGGY AZALEA (@IGGYAZALEA) December 23, 2017 Answer: “he was such a character”
Question: why will jordan feldstein be missed? Answer generated by SQuAD: “He was such a character ” Answer generated by TweetQA: “Jordan Feldstein passed”

Table 10. Example of TweetQA sentence relation and semantic understanding errors

Context: ”y’all are hating on nomaj but its lowkey growing on me” tria (@tonycest) November 4, 2015 Answer: “the hating, nomaj”
Question: what is growing on nomaj? Answer generated by SQuAD: “nomaj” Answer generated by TweetQA: “lowkey”

is given in Table 8, when the tweetQA model correctly understood the text but the answer was paraphrased. TweetQA also lags in understanding deep semantic, sentence relation and common sense problems. Example of that can be seen in Table 9 and 10.

5 Conclusion

We analyzed question answering derived from Twitter data and discussed the impact of data cleaning in social media for question answering. We mapped the problem into the extractive question answering problem to observe the results in a noisy social media context. We fine-tuned our transformer models on the SQUAD 2.0 baseline for all extractive question answering tasks, since SQUAD leads the way with a balanced question answering dataset.

Our results indicate that fine-tuned on extractive question answering setting TweetQA data can improve teh results and reduce significantly the gap between human evaluation results, when trained on ALBERT without hashtag and “at” sign.

However, it is not possible to reach the extractive upper bound by solely relying on the extractive setting, since social media text contains a lot of anomalies and can mitigate the evaluation scores.

We also conducted experiments with transformer models trained with TweetQA instance, which were the substring of the context. The results indicated a significant gap in the human evaluation and the achieved scores.

The error analysis of the mentioned models struggled in sentence relation, common sense and deep semantic questions. Both SQuAD and TweetQA trained models did well in understanding tweet specific attributes such as hashtags and “At” sign. In the extractive setting however, the informal structure of the tweets made a huge impact on the errors that were observed in the experiments.

Acknowledgment

The work was done with partial support from the Mexican Government through the grant A1-S-47854 of the CONACYT and grants 20211884 and 20211784 of the Secretaría de Investigación y Posgrado of the Instituto Politécnico Nacional, Mexico. The authors thank CONACYT for the computer resources provided through the INAOE Supercomputing Laboratory’s Deep Learning Platform for Language Technologies. Work was done while the third author was with the Instituto Politécnico Nacional, Centro de Investigación en Computación, Mexico.

References

1. **Chatterjee, A., Narahari, K. N., Joshi, M., Agrawal, P. (2019).** Semeval-2019 task 3: Emocontext contextual emotion detection in text. Proceedings of the 13th International Workshop on Semantic Evaluation, pp. 39–48.

2. **Choi, E., He, H., Iyyer, M., Yatskar, M., Yih, W.-t., Choi, Y., Liang, P., Zettlemoyer, L. (2018).** QuAC: Question answering in context. Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, pp. 2174–2184.

3. **Denkowski, M., Lavie, A. (2011).** Meteor 1.3: Automatic metric for reliable optimization and evaluation of machine translation systems. Proceedings of the Sixth Workshop on Statistical Machine Translation, Association for Computational Linguistics, Edinburgh, Scotland, pp. 85–91.

4. **Devlin, J., Chang, M.-W., Lee, K., Toutanova, K. (2018).** Bert: Pre-training of deep bidirectional transformers for language understanding. CoRR, Vol. abs/1810.04805.

5. **Dhingra, B., Liu, H., Yang, Z., Cohen, W., Salakhutdinov, R. (2017).** Gated-attention readers for text comprehension. Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Vancouver, Canada.

6. **Dunn, M., Sagun, L., Higgins, M., Guney, V., Cirik, V., Cho, K. (2017).** SearchQA: A new Q&A dataset augmented with context from a search engine. arXiv.

7. **Joshi, M., Chen, D., Liu, Y., Weld, D., Zettlemoyer, L., Levy, O. (2019).** SpanBERT: Improving pre-training by representing and predicting spans. arXiv.

8. **Kadlec, R., Schmid, M., Bajgar, O., Kleindienst, J. (2016).** Text understanding with the attention sum reader network. Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Berlin, Germany.

9. **Kocisky, T., Schwarz, J., Blunsom, P., Dyer, C., Hermann, K., Melis, G., Grefenstette, E. (2017).** The narrativeQA reading comprehension challenge. Transactions of the Association for Computational Linguistics.

10. **Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., Soricut, R. (2019).** ALBERT: A lite BERT for self-supervised learning of language representations. arXiv.

11. **Lin, C.-Y. (2004).** ROUGE: A package for automatic evaluation of summaries. Text Summarization Branches Out, Association for Computational Linguistics, pp. 74–81.

12. **Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V. (2019).** RoBERTa: A robustly optimized BERT pretraining approach. arXiv.

13. **Mandar, J., Eunsol, C., Daniel, S., Luke, Z. (2017).** TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. arXiv.

14. **Papineni, K., Roukos, S., Ward, T., Zhu, W.-J. (2002).** Bleu: A method for automatic evaluation of machine translation. Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, pp. 311–318.

15. **Rajpurkar, P., Jia, R., Liang, P. (2018).** Know what you don't know: Unanswerable questions for SQuAD. CoRR.

16. **Reddy, S., Chen, D., Manning, C. (2018).** CoQA: A conversational question answering challenge. Transactions of the Association for Computational Linguistics.

17. **Seo, M., Kembhavi, A., Farhadi, A., Hajishirzi, H. (2016).** Bidirectional attention flow for machine comprehension. arXiv.

18. **Song, L., Wang, Z., Hamza, W. (2017).** A unified query-based generative model for question generation and question answering. arXiv.

19. **Trischler, A., Wang, X., T. Yuan, Harris, J., Sordoni, A., Bachman, P., Suleman, K. (2016).** NewsQA: A machine comprehension dataset. CoRR.

20. **Voorhees, E., Tice, D. (2000).** Building a question answering test collection. Association for Computing Machinery, New York, NY, USA.

21. **Wang, W., Yang, N., Wei, F., Chang, B., Zhou, M. (2017).** Gated self-matching networks for reading comprehension and question answering. Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, volume 1, pp. 189–198.

22. **Xiong, W., Wu, J., Wang, H., Kulkarni, V., Yu, M., Guo, X., Chang, S., Wang, W. Y. (2019).** TweetQA: A social media focused question answering dataset.

Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics.

23. **Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R., Le, Q. V. (2019).** Xlnet: Generalized autoregressive pretraining for language understanding. arXiv.

24. **Yang, Z., Qi, P., Zhang, S., Bengio, Y., Cohen, W., Salakhutdinov, R., Manning, C. (2018).** HotpotQA: A dataset for diverse, explainable multi-hop question answering. Conference on Empirical Methods in Natural Language Processing (EMNLP).

25. **Yatskar, M. (2019).** A qualitative comparison of CoQA, SQuAD 2.0 and QuAC. Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Association for Computational Linguistics, Minneapolis, Minnesota.

26. **Zhang, Z., Yang, J., Zhao, H. (2020).** Retrospective reader for machine reading comprehension. arXiv.

Article received on 04/09/2020; accepted on 11/12/2020.
Corresponding author is Alexander Gelbukh.