УДК 004.852

AUTOMATIC QUESTION ANSWERING USING TOPIC MODELING IN PROGRAMMING DOMAIN

L. Rvanova (ITMO university)
Scientific supervisor – **PhD., Associate Professor S.V. Kovalchuk**
(ITMO university)

Introduction. Generative neural networks are currently widely used and are being actively researched. It is interesting to use generative neural systems in the task of automatically answering questions. Our task is to study the application of generative neural networks for automatic generation of Stack Overflow answers. The complexity of this task lies in the fact that in both answers and questions there are several domains at once: code, natural language, and images.

Currently, generative neural networks, such as GPT-3 (Generative Pre-Trained Transformer), are good at general questions, including some factual ones. T5 (Text-to-Text Transfer Transformer) achieved state-of-the-art result in several natural language processing tasks, including text generation. This is sequence-to-sequence transformer pre-trained on a large text corpus. Finally, nowadays we have ChatGPT (Chat Generative Pre-Trained Transformer) which handles multidomain responses to questions by being able to generate code along with natural language. Also, this solution has the ability to remember the context and correct errors. At the moment there is no article explaining how the solution works and there is no open source code. Unfortunately, we cannot be sure of the accuracy of the answers of ChatGPT.

We need to have some solution for automatically question answering in information technology, that can deal with several domains. The first step of our research – to determine how well classical text generation methods work on various topics in the natural language domain.

Main part. We used open sourced Stack Overflow dataset dumps. Our data has been filtered from inappropriate domains for this experiment, leaving only natural language. In experiments only newest questions over the past six months and approved or top-rated answers for them were used. For our experiments we conducted thematic modeling of questions. There are two types of thematic modeling: Latent Dirichlet Allocation and Correlated topic models. Also we used two types of texts for modeling: question titles and questions tags. Thematic modeling for the number of topics from 1 to 15 in order to find the optimal number of topics for coherence score was calculated. To decrease number of words in vocabulary lemmatized words were lemmatized. We used TF-IDF (term frequency-inverse document frequency) to remove stop-words. For topic modeling optimal number of topics — 8 for both of methods.

GPT-Neo (Chat Generative Pre-Trained Neo) was used for text generation. It is GPT-2 like model trained on the Pile dataset. It is transformer-based neural network trained on task of predicting next word in the sequence. GPT Neo uses local attention for every layer with window size of 256 tokens. We used 1.3 B configuration with 1.3 billion weights. In our experiments we used few-shot learning for inference. The perplexity fluctuations are higher than for topic modeling by question headings.

Despite the lower scores, topic modeling for question titles revealed clearer topics than modeling for tags. At the same time, differences in metrics are more significant for modeling by tags, although they are minor. The most visible are the differences in perplexity, for less specific topics the perplexity is lower, for more specialized topics it is higher.

Conclusion. GPT-Neo performs well even out of the box, showing good results in semantic similarity. However, for highly specialized topics, perplexity suffers. Perplexity also handles questions about software better than questions than questions about programming languages. In connection with these differences, it makes sense to retrain the model for each topic separately. It makes sense to continue experimenting with topic modeling, since tags are set by users and may not reflect the essence of the issue, just as headings may be worded incorrectly. This solution may be used for automatically question-answering related to programming.

Literature:

1. U. Arora, N. Goyal, A. Goel, N. Sachdeva and P. Kumaraguru Ask It Right! Identifying Low-Quality questions on Community Question Answering Services. // International Joint Conference on Neural Networks (IJCNN) — Padua, Italy — 2022 — pp. 1-8
2. Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, Dario Amodei Language Models are Few-Shot Learners // NIPS, 2020
3. Muller, Benjamin and Soldaini, Luca and Koncel-Kedziorski, Rik and Lind, Eric and Moschitti, Alessandro Cross-Lingual Open-Domain Question Answering with Answer Sentence Generation // Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), 2022, Association for Computational Linguistics, main.27", 337–35
4. Sid Black, Stella Biderman, Eric Hallahan, Quentin Anthony, Leo Gao, Laurence Golding, Horace He, Connor Leahy, Kyle McDonell, Jason Phang, Michael Pieler, USVSN Sai Prashanth, Shivanshu Purohit, Laria Reynolds, Jonathan Tow, Ben Wang, Samuel Weinbach GPT-NeoX-20B: An Open-Source Autoregressive Language Model, // BigScience (ACL) 2022