Automatic structuring of topics for natural language generation in community question answering in programming domain

Lyudmila Rvanova[1,2][0009−0007−7026−8829] and Sergey Kovalchuk[1][0000−0001−8828−4615]

[1] ITMO University, Saint Petersburg, Russia
[2] Artificial Intelligence Research Institute, Moscow, Russia
alfekka@itmo.ru, kovalchuk@itmo.ru

Abstract. The present article describes the methodology for the automatic generation of responses on Stack Overflow using GPT-Neo. Specifically, the formation of a dataset and the selection of appropriate samples for experimentation are expounded upon. Comparisons of the quality of generation for various topics, obtained using thematic modeling of the titles of questions and tags, were carried out. In the absence of consideration of the structures and themes of texts, it can be difficult to train models, so the question is being investigated whether thematic modeling of questions can help in solving the problem. Fine-tuning of GPT-neo for each topic is undertaken as a part of experimental process.

Keywords: Stack Overflow · Question answering · Text generation · Topic modeling

1 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 [1], are good at general questions, including some factual ones. GPT-3 is an autoregressive transformer model with 175 billion parameters. It based on GPT-2 [2] architecture including pre-normalization, reversible tokenization. This model is different from GPT-2 with their sparse attention patterns in the layers of transformer.
T5 made a breakthrough in multitasking. T5 achieved state-of-the-art result in several NLP tasks, including text generation. This is seq-to-seq transformer pre-trained on a large text corpus.
Finally, nowadays we have ChatGPT which handles multi-domain responses to questions by being able to generate code along with natural language. Also, this

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36021-3_33

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.
In March 2023, OpenAI released the GPT-4 [3] model. According to the OpenAI press release, GPT-4 scores 40% higher than the latest GPT-3 on internal adversarial factuality evaluations by OpenAI. Although this advantage is significant, the developers from Open AI confirm that GPT-4 does not completely solve the problem of generating inappropriate code and inaccurate information.
We're investigating how the topic of a question impacts answer quality. Our method uses thematic modeling and fine-tuning of models for each theme, improving accuracy of answers. By identifying relevant topics within a question, we can generate more helpful responses. It is also important to take into account structural differences in different texts within the same domain. For questions of different topics, different response structures are assumed, and if this is not taken into account, it can lead to problems in training the model.

2 Related works

The paper [4] explores the possibility of identifying low-quality questions on Stack Overflow for their automatic closure. Previous work has explored lexical, voice-based, style-based features. They also proposed a framework that collects semantic information about questions using transformers and explores information from tags and questions using a convolutional graph neural network. This method beat the stat-of-the-art solutions.
In paper [1] few-shot learning for generative neural networks was studied. Most of the SOTA results in text generation are obtained by retraining and fine tuning on thousands and hundreds of thousands of examples. Training on a small number of examples usually cannot beat the results of such a tuning, but when scaled and using a large language model, it can be successful. In this work, authors used GPT-3 with 175 billion parameters.
Researches in this [5] article states that Open-Domain Question Answering achieved good results with combining document-level retrievers with text generation. This approach also called GENQA can affectively answer both factoid and non-factoid questions. They introduce GEN-TYDIQA, an extension of the TyDiQA dataset with well-formed and complete answers for Arabic, Bengali, English, Japanese, and Russian. They translate question to one of appropriate languages, uses retriever, monolingual answer detection and aggreagtion of answer. After that they uses cross-lingual GenQA.

3 Data

We used open sourced Stack Overflow dataset dumps. There are only two files that we used – datadump of Stack Overflow posts dated December 7, 2022 and dump of Stack Overflow comments dated December 6, 2022. The oldest entries

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36021-3_33

date back to 2008. XML files have been converted to a suitable format for us. There are 57721551 records for file with posts and 86754114 for comments.
Our data has been filtered from unnapropriate domains for this experiment, leaving only natural language. For this we simply deleted answers and questions with tags of code and pictures. In our experiments we used only newest questions over the past six months and approved or top-rated answers for them. As the result, we have 25668 questions with answers.

4 Methods

4.1 Thematic modeling

For our experiments we conducted thematic modeling of questions. There are two types of thematic modeling:

1. **LDA** [6] (Latent Dirichlet Allocation) — generative probabilistic method. We used Tf-Idf to delete stop-words.
2. **CTM** [7] (Correlated topic models) — hierarchical model that allows us to use the correlation of latent topics. It is extension of LDA.

Also we used two types of texts for modeling: question titles and questions tags. We calculated thematic modeling for the number of topics from 1 to 15 in order to find the optimal number of topics for coherence score. To decrease number of words in vocabulary we lemmatized words. Stopwords were removed as common practice in topic modeling. We used TF-IDF for stopwords removing. For LDA we used n-grams with sizes 2 and 3. For CTM we used combination of Bag of Words and BERT base cased embeddings.
For thematic modeling we used short headings of questions.

4.2 Text generation

In our experiments we used GPT-Neo [8] for text generation. It's 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.

5 Results

For the general dataset, we have the following metrics in the case of generation by GPT-Neo: ROUGE1, ROUGE2, ROUGEL, ROUGELsum, GoogleBLEU, average perplexity, cosine similarity.

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36021-3_33

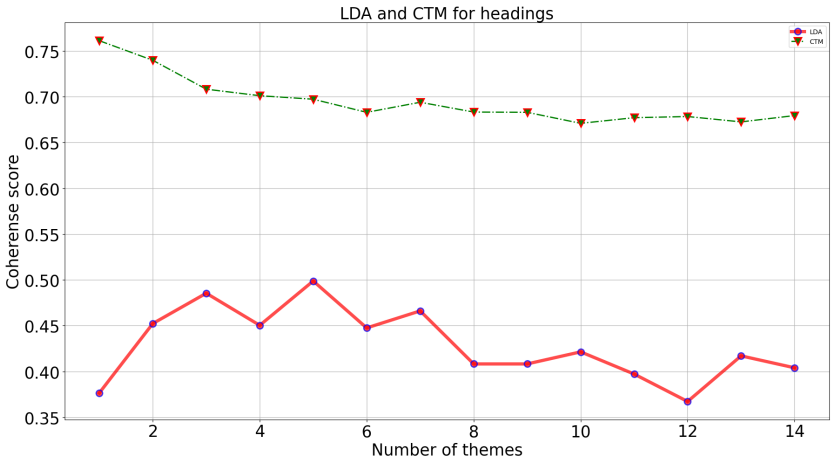

Fig. 1. Thematic modeling with LDA and CTM for headings of questions.

Headings of questions. Optimal number of topics for CTM is 8, for LDA is 8. In this case coherence score were higher for CTM modeling than for LDA modeling (Fig. 1).

We have next keywords for topics in case of CTM modeling and 8 topics:

1. time, data, database, order, case, unrestand, process – general questions and data
2. version, token, command, project, error, build, package – system administration questions
3. description, norefferimg, alterner image, tr thread, styletextalign, table div, tr tbody – frontend
4. azure, access, token, server, api, client, application – backend and database questions
5. cloudwatch, cloudflare, attachments, collections, sftp, ms teams, kubernetes cluster – DevOps
6. english, inevitably, bcnf, pem string, justify, direct channel, subscrible channel
7. fiscal, deeply, perm strong what, immensely, source directory, looped, alt unloaded
8. classes methods, importing, jsonincludeproperties, java objectives, immensely, taget build, puzzled – java

The last three topics overlap a lot.

For each topic we sampled 2000 questions and generated answers with GPT-Neo. Results are below (Table 1). We used cosine similarity to find out similarity between texts. To measure cosinus similarity we used SentenceTransformers [9] embeddings.

In this case, there are slight fluctuations in perplexity – for questions on Java, its

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36021-3_33

Table 1. Results of GPT-Neo few-shot text generation for different topics modeled using headings.

	Topic 1	Topic 2	Topic 3	Topic 4	Topic 5	Topic 6	Topic 7	Topic 8
ROUGE1	0.16	0.16	0.17	0.15	0.15	0.17	0.15	0.16
ROUGE2	0.02	0.02	0.02	0.03	0.01	0.02	0.01	0.02
ROUGEL	0.11	0.11	0.11	0.10	0.10	0.11	0.10	0.10
ROUGELsum	0.11	0.11	0.11	0.10	0.10	0.11	0.10	0.10
Goggle BLEU	0.02	0.02	0.03	0.02	0.02	0.03	0.02	0.02
Avg perplexity	67.04	66.69	65.36	71.09	67.97	54.76	61.78	65.41
Cos similarity	0.87	0.88	0.89	0.88	0.88	0.89	0.87	0.87

Table 2. Results of GPT-Neo fine-tuned text generation for different topics modeled using headings.

	Topic 1	Topic 2	Topic 3	Topic 4	Topic 5	Topic 6	Topic 7	Topic 8
ROUGE1	0.19	0.18	0.18	0.17	0.18	0.19	0.19	0.17
ROUGE2	0.02	0.02	0.02	0.02	0.02	0.03	0.03	0.02
ROUGEL	0.12	0.13	0.12	0.12	0.12	0.13	0.13	0.11
ROUGELsum	0.14	0.14	0.14	0.13	0.14	0.15	0.15	0.13
Goggle BLEU	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04
Avg perplexity	42.41	53.18	46.76	50.43	55.26	44.00	56.23	66.47
Cos similarity	0.91	0.91	0.90	0.91	0.91	0.92	0.91	0.91

value is better than for system administration, databases and frontend. Other metrics differ less significantly. It makes sense to do another experiment with topic modeling – this time we'll take question tags and run the simulation in the same way.

After experiments with inference of GPT-Neo we fine-tuned models on StackOverflow dataset. For each topic we conducted fine-tuning separately with data that was determined as beloning to this topic. We end up with a similar improvement in metrics for all topics, but the responses become more relevant to the topics. When fine-tuning on the basis of all data that is not grouped by topic, the metrics also grow, but the answers do not correspond to the topics of the questions (Table 2).

Practically everywhere except for the last topic, perplexity has decreased. For all topics except the third one, other metrics improved significantly, by 10-40% . The seventh topic showed the best increase in metrics.

Tags of questions. Optimal number for both of CTM and LDA is 8. For tags coherence score is higher for LDA (Fig. 2)

We have next keywords for topics in case of CTM modeling and 8 topics:

1. android, python selemium, react js, excel formula, html css, my sql, asp net deployment
2. visual studio, apache flink streaming, power bidax, python django, google cloud platform, snowflake clout dataplatform, heroku

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36021-3_33

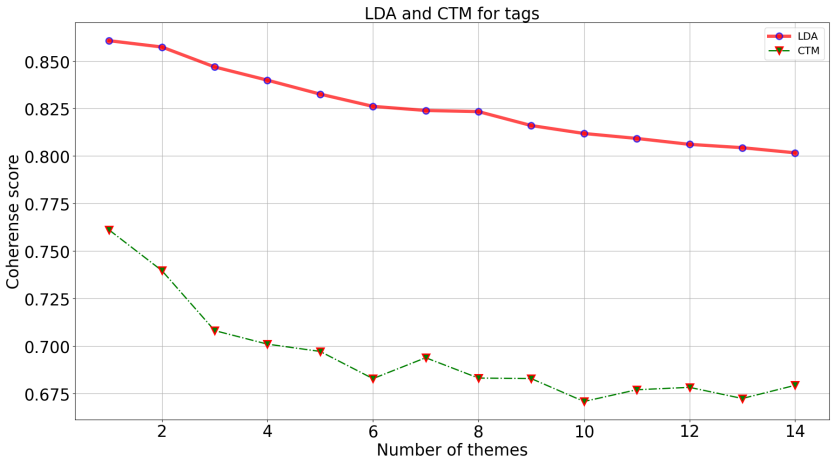

Fig. 2. Thematic modeling with LDA and CTM for tags of questions.

3. git, github, microsoft teams, sql web service, pycharm, kubernetes, angular, res
4. visual studio code, java android, algorithm, excel, azure, python 3x, excel pivot table
5. flutter dart, mongo db, azure devops, docker, bash, logging, apache kafka consume api
6. unity 3d, c, java androin kotlin, camunda, python algorithm, flutter, java mysql spring
7. python, powerbi, apache kafka, java, td engine, spring, apache spark,apache spark
8. javascript, react, frontend, github, android studio, postgre sql, azure devops

The last three topics overlap a lot.
For each topic we sampled 2000 questions and generated answers with GPT-Neo (Table 3). For tags we also conducted fine-tuning (Table 4).

Table 3. Results of GPT-Neo few-shot text generation for different topics modeled using tags of questions.

	Topic 1	Topic 2	Topic 3	Topic 4	Topic 5	Topic 6	Topic 7	Topic 8
ROUGE1	0.16	0.15	0.15	0.15	0.15	0.17	0.17	0.16
ROUGE2	0.02	0.02	0.01	0.01	0.01	0.02	0.02	0.02
ROUGEL	0.10	0.09	0.09	0.10	0.09	0.11	0.10	0.11
ROUGELsum	0.10	0.09	0.09	0.10	0.09	0.11	0.10	0.11
Goggle BLEU	0.02	0.02	0.02	0.03	0.02	0.03	0.02	0.02
Avg perplexity	63.22	43.40	49.15	48.43	54.76	88.7	84.32	65.52
Cos similarity	0.87	0.86	0.88	0.89	0.89	0.89	0.87	0.87

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36021-3_33

Table 4. Results of GPT-Neo fine-tuned text generation for different topics modeled using tags of questions.

	Topic 1	Topic 2	Topic 3	Topic 4	Topic 5	Topic 6	Topic 7	Topic 8
ROUGE1	0.19	0.17	0.18	0.19	0.19	0.18	0.18	0.19
ROUGE2	0.03	0.02	0.02	0.03	0.02	0.02	0.03	0.02
ROUGEL	0.13	0.12	0.12	0.13	0.12	0.12	0.13	0.12
ROUGELsum	0.15	0.13	0.14	0.15	0.14	0.14	0.14	0.15
Goggle BLEU	0.04	0.03	0.04	0.04	0.04	0.04	0.04	0.04
Avg perplexity	50.55	66.90	48.35	47.58	60.26	50.59	44.89	48.09
Cos similarity	0.91	0.91	0.92	0.92	0.92	0.92	0.91	0.91

In this cased perplexity decreased for every topic except the second. For every topic metrics improved, especially for the fourth theme.
Despite the large overlap of the last three topics, there is a significant difference in perplexity between them. Topics 2, 3 and 4 showed the best perplexity values, but showed a relatively low ROUGE values.
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.

6 Conclusion

GPT-Neo performs well even out of the box, showing good results in semantic similarity. However, for highly specialized topics, perplexity suffers. She also handles questions about software better than questions than questions about programming languages.
Additional training for each topic separately showed an improvement in quality. The answers to the questions are more qualitatively related to the topic of the question in the case of training separately than in the case of additional training on all the data mixed.
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.

Acknowledgments

This research is supported by Russian Scientific Foundation and Saint Petersburg Scientific Foundation, grant No. 23-28-10069 "Forecasting social well-being in order to optimize the functioning of the urban digital services ecosystem in St. Petersburg" (https://rscf.ru/project/23-28-10069/).

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36021-3_33

References

1. 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, In: H. Larochelle and M. Ranzato and R. Hadsell and M.F. Balcan and H. Lin (eds.) Advances Neural Information Processing Systems, vol. 33, pp. 1877–1901, Curran Associates, Inc. (2020)
2. Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever: Language Models are Unsupervised Multitask Learners, Technical report, OpenAi (2019)
3. OpenAI: GPT-4 Technical Report, eprint arXiv:2303.08774 (2023)
4. U. Arora, N. Goyal, A. Goel, N. Sachdeva and P. Kumaraguru: Ask It Right! Identifying Low-Quality questions on Community Question Answering Services. In: International Joint Conference on Neural Networks (IJCNN) pp. 1-8, Padua, Italy (2022), doi: 10.1109/IJCNN55064.2022.9892454.
5. 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, In: 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, vol. 1, pp. 337–353, Association for Computational Linguistics (2022),
6. David M. Blei, Andrew Y. Ng, Michael I.: Latent Dirichlet Allocation, In: Jordan Journal of Machine Learning Research 3, pp. 993–1022 (2003)
7. David M. Blei, John D. Lafferty: Correlated Topic Models, In: Y. Weiss and B. Schölkopf and J. Platt (eds.) Advances in Neural Information Processing Systems (NIPS 2005), vol. 18, pp. 147-154, MIT Press (2005)
8. Sidney 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, In: Proceedings of BigScience Episode #5 – Workshop on Challenges & Perspectives in Creating Large Language Models, pp. 95–136, Association for Computational Linguistics (2022), doi:10.18653/v1/2022.bigscience-1.9
9. Nils Reimers and Iryna Gurevych: Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks, In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pp. 3982–3992, Association for Computational Linguistics, doi:10.18653/v1/D19-1410

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36021-3_33