NLP Course Template

Denis Pikhenko

December 2021

Abstract

This document will provide you with guidelines for your project final report. You will learn how to structure the report and present your results. Please provide a link to your project code right here: `https://github.com/yournickname/your-project-name`.

1 Introduction

In this article, I am developing a question answering (QA) system that will be able to answer those questions student's ITMO University that do not require an individual approach.

QA systems are used inside chatbots frequently [Sharma and Gupta, 2018]. They are used as a support service to answer questions that customers ask. Such an example could be the ALICE bot. It uses an artificial intelligence markup language (AIML) which applies the technique of pattern recognition or pattern matching [Marietto et al., 2013].Other bots such as Ichat Indonesia and Veronica use natural language processing (NLP) and rule-based methods, however it makes them less flexible. It is also necessary to understand that creating a QA system based on rules and patterns is a difficult task [Shang et al., 2015].However, there are other approaches that require not rules but data (questions and answers) with which we can develop a QA system [Luan et al., 2016]. One of these approaches is the sequence-to-sequence approach (SEQSEQ). With the help of it was shown that pre-trained generative seq2seq models can achieve high performance in QA tasks [Izacard and Grave, 2020, Roberts et al., 2020].Based on this data, I used the generative T5 model [Raffel et al., 2019] to generate answers to the question from my dataset.

1.1 Team

The author of the project is **Denis Pikhenko**, Saint Petersburg National Research University of Information Technologies, Mechanics and Optics master student in a data analytics.

2 Related Work

Answering questions remains one of the most difficult tasks related to artificial intelligence.No one question answering system can not still answer questions like an ordinary person. However to achieve it people create different QA datasets for example: SQuAD [Rajpurkar et al., 2016], WikiQA [Yang et al., 2015], CNN/DailyMail [Hermann et al., 2015], MS MARCO [Bajaj et al., 2016], TriviaQA [Joshi et al., 2017].

There are different approaches related to question answering. They could be divided into: extractive QA, abstract QA and closed-book QA. The first two approaches are often complemented by a selective module (retriever). Basically, the complete quality control pipeline for extractive QA and abstract QA looks like this: from a large volume of documents (for example, Wikipedia), the retriever selects a much smaller subset. Then the reader checks them and either selects a range of sequential tokens (for QA extraction), or forms a response using the selected passages as input in a sequence-to-sequence setting (for abstract QA).Closed-book QA is the most recent trend, which is made possible thanks to the availability of extremely large language models, such as GPT-3 [Brown et al., 2020],T5 [Raffel et al., 2019] and mT5 [Xue et al., 2020]. These generative solutions are able to answer questions using only the world knowledge preserved in the model itself. They reflect the human ability of answering a question without consulting any external knowledge sources.

3 Model Description

In this work I used two models: BERT [Devlin et al., 2018] and T5 [Raffel et al., 2019].

BERT is necessary for the classification of questions. T5 model is needed to generate answers to these questions. It should be noted that it is quite difficult to train the T5 and BERT models from scratch, since this requires a lot of time and expensive equipment. In my work, I used transfer learning. Transfer learning is an approach that consists in the fact that the model is pre-trained on big data using powerful computers (by other people) and then it can be fine tuned to its specific task [Sun et al., 2019, Radford et al., 2018].Let's consider the architecture of the BERT and T5 models shortly.

3.1 T5 model

T5 is a language model, the difference of which is the use of a single text-to-text format for NLP tasks. This approach looks quite natural for tasks in which something needs to generate (for example, a machine translation task). The basic architecture of T5 with which it converts some sentences into others is the encoder-decoder, which was proposed by [Vaswani et al., 2017].The main advantage of this approach is that for each task in the learning process, the model concentrates on the same goal (teacher-forced maximumlikelihood), that means that one set of hyperparameters can be used for fine-tuning in other tasks. What kind of task is used for T5 pre-training? Such a task is the prediction of

masked tokens. Tokens are fed into the model sequentially, some of which are replaced by masks. The main task of the model is to predict the masked token correctly.

3.2 BERT model

The BERT model is a multi-layer bidirectional Transformer encoder based on the Transformer model [Vaswani et al., 2017].The input representation is a concatenation of embeddings from dictionary of wordpieces [Wu et al., 2016], positional embeddings, and segment embeddings. Initially, BERT is trained on two tasks: masked token prediction and classification task. What exactly is classified? The model determines one sentence is included in another or not.A special token ([CLS]) is used for this classification exactly.It is always at the beginning of the first sentence. Also, a token ([SEP]) is used to separate sentences, which is located at the end of the first and second sentences. Thus, the pre-trained BERT model has a powerful context-dependent representation of sentences and can be used for different tasks. In my case, this is a classification task.

3.3 The main algorithm

All the answers from my dataset can be divided into three groups - actions, facts and references. Actions do not imply an automatic response - the operator's help is needed here. Facts are detailed answers to a particular situation in this case, a person is also required. In the action group, I include answers in which there is a verb of the perfect form of the past tense.

I guess the answers for actions and facts are long and detailed, and the answers for references are short. Let's make an exploratory data analysis (EDA) on the length of answers and questions.

I highlighted the references by the length of the answer - no more than 40 characters (A9). This is about 1000 records or 25% of the dataset. Questions with a short answer (in line length) do not differ in any way from other questions on Fig. 1.

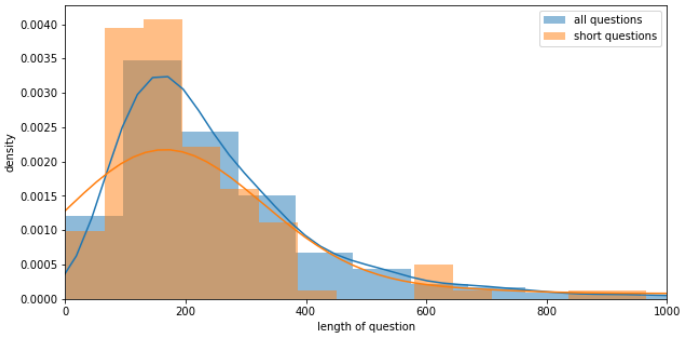

Figure 1: distribution of the length of questions.

Then I used the resulting class label to classify the questions using BERT [González-Carvajal and Garrido-Merchán, 2020]. The BERT learns to classify the length of answers by questions. As a result, the accuracy of the model on validation is 85%.

The final algorithm for questions answering:

1. Classify the question.
2. If the question is an action or a fact, then a standard answer is given: "The question has been passed to the operator".
3. If the question is a reference, then the answer is generated by calling the T51 model.

4 Dataset

I used question-answer pairs as a dataset. All questions and answers are written in Russian. The questions were written by students about a certain online course they took at ITMO University. The answers to each question were written by people from the support service. Each question is matched with the correct answer.

The dataset was split into train, validation and test sets before the model training. Tab. 1 shows the dataset statistics.

	Train	Valid	Test
QA pairs	3,320	712	712
Tokens in answers	67,884	14,271	14,692
Tokens in questions	147,717	31,855	31,027
Total tokens	215,601	46,126	45,719

Table 1: Statistics of the dataset.

5 Experiments

5.1 Metrics

I used accuracy as a score of the classification.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

where TP - True positive (a test result that correctly indicates the presence of a condition or characteristic);

FP - False positive (a test result which wrongly indicates that a particular condition or attribute is present);

TN - True negative (a test result that correctly indicates the absence of a condition or characteristic);

FN - False negative (a test result which wrongly indicates that a particular condition or attribute is absent);

As an estimate of the generative model T51, I used the human evaluation. This is due to the fact that automatic evaluation of such models is quite difficult and its application does not always show the real quality of the model. The score was given on a five-point scale, where 5 is the best result.

5.2 Experiment Setup

The model BERT was fine tuned in Google Colaboratory with the Tesla P100 GPU. The model T5 was fine tuned in CPU Intel Core i5-5200U. The dataset was divided into a training (70% of the dataset), validation (15% of the dataset) and test (15% of the dataset) sample. The PyTorch library was used to train the models.

The configuration of the BERT based model is the following. Ithas 12 layers, the hidden size equal to 768, the number of self-attention heads equal to 12 and total parameters equal to 110 million. The model ran for 11 epochs. The learning rate is 2e-5.

The configuration of the T51 based model is the following. Both the encoder and decoder consist of 12 blocks (each block comprising self-attention, optional encoder-decoder attention, and a feed-forward network). The feed-forward networks in each block consist of a dense layer with an output dimensionality of dff = 3072 followed by a ReLU nonlinearity and another dense layer. The "key" and "value" matrices of all attention mechanisms have an inner dimensionality of dkv = 64 and all attention mechanisms have 12 heads. All other sub-layers and embeddings have a dimensionality of dmodel = 768. In total, this results in a model with about 220 million parameters. The model ran for 6 epochs.The learning rate is 1e-5.

5.3 Baselines

For classification, I used a model that consists of the BERT whose outputs are directed to the Dropout layer with the parameter p=0.2 and then to the Linear layer. The outputs of this layer give us the necessary classes. This model was fine tuned on the base « LaBSE-en-ru» model which was pre-trained in Russian and English texts. For generation answers I used the model which was pre-trained in Russian texts. It is called «rut5-base-multitask».All models I took from the HuggingFace Transformers library. The fine tuning was happened on my own dataset.

6 Results

The results of the BERT classification model and the generative T5 model are presented in Table 2.

Model	Accuracy	Human evaluation
BERT	0.85	-
T51	-	4

Table 2: Results on the test dataset.

The T51 model did not get the maximum score because it generates incorrect answers sometimes . Often these answers are related to specific dates and times. In other cases, it generates correct answers frequently.

7 Conclusion

In the course of the research, a system was created consisting of a BERT-based classifier and a T5-based response generator, which in the future can be used inside a chatbot. It should reduce the burden on the university's support service. At the moment, the task related to the answers to the references questions is almost solved. In the future, it is planned to make a detailed study of the remaining questions and highlight the most common ones, after which they will be given ready-made answers to them. It will allow the system to cover a larger number of issues.

References

[Bajaj et al., 2016] Bajaj, P., Campos, D., Craswell, N., Deng, L., Gao, J., Liu, X., Majumder, R., McNamara, A., Mitra, B., Nguyen, T., et al. (2016). Ms marco: A human generated machine reading comprehension dataset. *arXiv preprint arXiv:1611.09268*.

[Brown et al., 2020] Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. (2020). Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

[Devlin et al., 2018] Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

[González-Carvajal and Garrido-Merchán, 2020] González-Carvajal, S. and Garrido-Merchán, E. (2020). Comparing bert against traditional machine learning text classification. arxiv prepr. *arXiv preprint arXiv:2005.13012*.

[Hermann et al., 2015] Hermann, K. M., Kocisky, T., Grefenstette, E., Espeholt, L., Kay, W., Suleyman, M., and Blunsom, P. (2015). Teaching machines to read and comprehend. *Advances in neural information processing systems*, 28:1693–1701.

[Izacard and Grave, 2020] Izacard, G. and Grave, E. (2020). Leveraging passage retrieval with generative models for open domain question answering. *arXiv preprint arXiv:2007.01282*.

[Joshi et al., 2017] Joshi, M., Choi, E., Weld, D. S., and Zettlemoyer, L. (2017). Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. *arXiv preprint arXiv:1705.03551*.

[Luan et al., 2016] Luan, Y., Ji, Y., and Ostendorf, M. (2016). Lstm based conversation models. *arXiv preprint arXiv:1603.09457*.

[Marietto et al., 2013] Marietto, M. d. G. B., de Aguiar, R. V., Barbosa, G. d. O., Botelho, W. T., Pimentel, E., França, R. d. S., and da Silva, V. L. (2013). Artificial intelligence markup language: a brief tutorial. *arXiv preprint arXiv:1307.3091*.

[Radford et al., 2018] Radford, A., Narasimhan, K., Salimans, T., and Sutskever, I. (2018). Improving language understanding by generative pre-training.

[Raffel et al., 2019] Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. (2019). Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*.

[Rajpurkar et al., 2016] Rajpurkar, P., Zhang, J., Lopyrev, K., and Liang, P. (2016). Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*.

[Roberts et al., 2020] Roberts, A., Raffel, C., and Shazeer, N. (2020). How much knowledge can you pack into the parameters of a language model? *arXiv preprint arXiv:2002.08910*.

[Shang et al., 2015] Shang, L., Lu, Z., and Li, H. (2015). Neural responding machine for short-text conversation. *arXiv preprint arXiv:1503.02364*.

[Sharma and Gupta, 2018] Sharma, Y. and Gupta, S. (2018). Deep learning approaches for question answering system. *Procedia computer science*, 132:785–794.

[Sun et al., 2019] Sun, C., Qiu, X., Xu, Y., and Huang, X. (2019). How to fine-tune bert for text classification? In *China National Conference on Chinese Computational Linguistics*, pages 194–206. Springer.

[Vaswani et al., 2017] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

[Wu et al., 2016] Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., et al. (2016). Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

[Xue et al., 2020] Xue, L., Constant, N., Roberts, A., Kale, M., Al-Rfou, R., Siddhant, A., Barua, A., and Raffel, C. (2020). mt5: A massively multilingual pre-trained text-to-text transformer. *arXiv preprint arXiv:2010.11934*.

[Yang et al., 2015] Yang, Y., Yih, W.-t., and Meek, C. (2015). Wikiqa: A challenge dataset for open-domain question answering. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pages 2013–2018.