**EXPLANATIONS AND COMMENTS ON THE TASK.**

The task of generating logical consequences from a given premise (NLI - Natural Language Inference) is complex for the following reasons:

1. Generating a full logical consequence from an arbitrary statement is not a task of formal logic and requires, at a minimum, what is known as language understanding (NLU), common sense, the ability to reason logically, and knowledge about the surrounding world for high-level generation.

2. It is difficult to assess the quality of generation because, similar to the task of abstract summarization, the generated text of logical consequence may not contain any words from the original premise or may contain them but be meaningless.

3. In this particular problem setting, there is very little data - approximately 1,500 examples of pairs: premise - logical consequence.

Based on this, I can suggest **five** variants for solutions:

**Method 1**. A general solution to the problem of generating logical consequences - this method is likely to be the SOTA (State Of The Art) approach in the near future, while other methods are merely attempts to approximate it.

As my experience in researching LLMs (Large Language Models) within my master's thesis: LLMs. Learning and Reasoning at the Inference Stage, and experience in conducting research for writing papers on LLMs suggest: the task of generating a logical consequence is, in fact, the very Reasoning that was the subject of my study.

Here is what I can say about it:

The ability of Reasoning, that is, to extract from language and form reasoning patterns on the train and use them on inference, starts to significantly manifest in LLMs of size from 30-40B parameters. As the number of parameters increases, the quality of reasoning improves. (The dependency of quality on the number of parameters is clearly nonlinear, but there are no detailed studies on this topic. However, I conducted a general analysis based on many sources, the main one being the article Emergent Abilities of Large Language Models(LLMs) - <https://arxiv.org/abs/2206.07682>).

The main problem with training such a model is likely not in the size of the parameters and computational resources, but in the data, as collecting a sufficient dataset while limiting oneself to the Russian corpus is practically impossible. According to research in Google's article about Chinchilla, to achieve high quality model performance, 20 times more data is needed for training than the number of parameters – for example, for a model with 70B parameters, approximately 1.4T training tokens are required. However, the problem has a solution, as noted in recent research and as evident even in my solution to the test task. The multilingual model mT5, trained on Persian (Farsi) classification to determine whether the proposed hypothesis is a logical consequence, successfully copes with a similar task in Russian. That is, LLMs are capable of using skills in a cross-lingual manner. This means that a large Russian LLM should be trained on data in any available languages, further fine-tuning it to high-quality Russian on a specialized Russian corpus.

Following the understanding that the task of generation can only be fundamentally solved by using an advanced LLM, as an illustration, I generated logical consequences for test.csv using GPT 4.0.

Here is a link to the chat with the illustration of generating logical consequences (the second part of the page, in the first part the model intensively corrects the grammar of the request):

<https://chat.openai.com/share/a39aa360-5cc8-4ba5-b0d1-55940571cd8f>

Pay attention to the quality of generation.

The level is practically indistinguishable from that of an educated, rational-thinking person.

I believe that this approach is the future, and in the near future at that.

**Method 2.** Using only small available models (Huggingface), limited resources (Google Colab), and a small amount of data (in the 3 files: train, val, test). This method is the main one for completing the test task, without delving into various nuances of the question. The code in Jupyter Notebook is provided as part of this method of solution. In this case, I was more focused on demonstrating code using PEFT / LoRA, rather than achieving a high-quality result, which is difficult to obtain with this problem setting.

Method 3. Similar to method number 2, but with a significantly larger dataset, which can be generated using a large LLM. I created a demonstration with pairs: premise-logical consequence. The link is below. Generated using GPT4. I believe that a dataset of similar quality, with 100 thousand entries or more, can be used both for training LoRA with the number of trainable parameters around 20-30% of the size of the base model (which should preferably be as large as possible due to the complexity of the task), and simply to fine-tune the entire model - using conventional fine-tuning. Link to the chat with the generation of illustrative pairs: <https://chat.openai.com/share/2f899c0e-3991-48d6-af0d-dc0f3c97ee8a>

**Method 4.** Train the model on a large English-language dataset – the Stanford NLI dataset, which contains 570 thousand manually labeled pairs. Based on considerations about the cross-lingual nature of LLM skills, this should work on other languages known to the model, including Russian. Given the significant volume of the SNLI dataset, it's better to train a sufficiently large model and through fine-tuning of the model itself. Although in theory, one could try LoRA as well.

**Method 5.** Distillation from a large model into a smaller one specifically for the task of generating logical consequence. However, here it is necessary to check how high the quality of the result will be, as it's possible that the quality could significantly deteriorate. Predicting how such an advanced ability as the generation of logical consequence will distill without experiments is difficult.

One thing is clear in advance – the model being trained should be sufficiently large, already capable of independently using reasoning patterns, but not as titanic as the teacher model, generating reasoning at an expert level.