Hey, I’m Yuliya. I really like to work with data. I also like to write concise and clean code. My interest in software development and data analysis led me to machine learning, neural networks and everything related to this field. Here are my answers for test!

Part 1. Theory

1. What are the main problems of modern NLP and NLU?

Even though some NLP tasks like POS tagging are considered to be solved, NLU tasks remain open.
There are two classes of problem: data-related and understanding-related issues.
a) To define a problem properly it’s necessary to decide what kind of data and how much of it we need. Adding new documents to a dataset can lead to more variability. In some area (like low-resource languages) it is impossible to get more data. But to get the data is not enough, we need to build the efficient and clean dataset. The data can be highly complex, it can include tables, graphics, notations, page breaks and so on, which need to be appropriately processed. Also we need to define evaluation procedures to appropriately measure our progress towards a concrete goal.
Another problem is that RNN-based models don’t work well with large documents because of gradient vanishing problem that don’t let to catch long distance dependencies.

b) A system has to be able to understand context to solve such problems as:
- multiple meaning of a word,
- synonymy (little and small can by sinonyms, but“little brother” can mean “yonger brother” even if he is 60 years old).
- finding all expressions that refer to the same entity (coreference)
Also the pragmatic interpretation or understanding what the text is trying to achieve (to inform, to command, to make fun or something else) is open problem.

2.Which libraries would you pick to use for the following cases and why (all problems should be solved for the Russian)

I’d use Huggingface Transformer package and BERT with pretrained for Russian RuBERT:
BertForSequenceClassification for sentiment analysis
BertForMultipleChoice for multi-label classification
BertForTokenClassification for NER

Why? Because it’s state-of-the-art and I’m interested to try it!

For POS-tagging I would use pymorphy2 because it’s very easy to use.

For dependency parsing I would use UDPipe 2.0 with SynTagRus treebank. Why? Because UDPipe is a fast pipeline that includes tokenization, tagging, lemmatization and dependency parsing all in one.

3. How would you evaluate a classification model, which metrics would you use?

For classification problems I would use precision or/and recall, or f1 score which is the combination of recall and precision. For a multiclass problem precision and recall calculate by reducing it to K binary classification problems, computing K precision and recall and averaging them.

4. Main pipeline for the text pre-processing.

1) Tokenization - Splitting to tokens (sentences, words)

2) Cleaning - Removing special characters, punctuations, stopwords

3) Normalization - Convertion of any non-text information (dates, nmbers, currency) into textual equivalent.

4) Stemming - Removing words prefixes and suffixes

5) Lemmatization - Casting words to their root forms.

5. Microservices or monoliths? Why.

In my opinion the best option is hybrid, where 90-95% is monolith and 5-10% is microservices. Distributed systems are more difficult to program, so you need to take into account all possible errors that can happen during connection between components and figure out how to solve them. In distributed systems it is more difficult to maintain strict consistency, remote calls are slower, and a large number of programming languages, development environments, and technologies require more specialists to cover the entire stack. Therefore, it is better to separate only large logically isolated fragments of the core into microservices to increase performance significantly.

6. Describe the hardest programming task you've been facing with. It's not necessarily ML
task, could be just a programming. Why this task was hard to accomplish? What was your solution for the task? Can you share a github project?

I worked with DWH. I designed some reports for a bank. I had to gather information from lots of sources, from different tables of different databases. I was really hard because data was noisy and it wasn't clear where to get necessary information. I had to spend lots of time talking with bank representatives from different departments, mining and cleaning data.
I can share my project of prediction price of used cars.
https://github.com/yulits/Test_task_nlp_watchout/blob/master/project_used_cars_yuliya_klimushina_from_repo_eng.ipynb

7. Did you work with VCS? Which one?

I well-versed in git. I can create a new repository, clone, push, pull, make a new branch, merge branches, make pull request and etc

8. Did you work with Github Actions?

0

9. How familiar are you with Docker and other orchestration tools?

In theory. Never used but don’t think it’s hard to learn

10. What is ed25519 and why is it concerning to be better than ecdsa?

Ed25519 is the most recommended cryptographic algorithm available today.It is used for SSH key. It offers a better security hicompared to ECDSA. There is a flaw in the way random numbers are used in ECDSA

11. Do you have any experience in data mining?

Yes! During my work on DWH

Part 2. Practice

Practice is here:
https://github.com/yulits/Test_task_nlp_watchout/blob/master/test_task_nlp_watchout_practice.ipynb