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

* Synonymy

* Ambiguity

* Personality, intention, emotions, and style

* Low-resource datasets for rare languages

* Coreference. The process of finding all expressions that refer to

the same entity in a text is called coreference resolution.

2.Which libraries would you pick to use for the following cases and

why (all problems should be solved for the Russian)

- Sentiment analysis. NLTK is easy to use and has useful base of documentation

- Multi-label classification. Sklearn has a lot of documentation and a big community.

I don't choose DeepPavlov for my problem beacuse it's require a good GPU

- Dependency parsing. DeepPavlov

- POS-tagging. DeepPavlov

- NER. DeepPavlov has good scores and specialize in Russian

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

For the lenta dataset I used f1-score, precision and recall(precision and recall depends on the goal). Accuracy isn't good metric if the dataset is unbalanced. Also we can use logistic loss and ROC_AUC metrics.

4.Main pipeline for the text pre-processing.

Usual pipeline for the text pre-processing consists:

- Clean data removing special characters: keep only what can be useful for the context;

- Tokenization

- POS tagging. Tagging can help to filter unwanted POS's (keep adjectives, adverbs, verbs and nouns).

- Lemmatizing: this will allow us to reduce our vocabulary. Use stemming if no need for precision and speed is preferred.

- Remove stopwords.

5.Microservices or monoliths? Why.

I would choose a combination of these two architectures, that is, hybrid. Which would be able to combine advantages and remove disadvantages. But if I only need to choose one of them, then I like the monolith architecture.

For me as an intern, monolithic architecture is more suitable.

- Monoliths is suitable for a small project.

- Everything is on one project, centralized management => easy to dev.

- Easy for debugging and testing

- Easy to deploy and improve.

- Most junior engineer has enough skills to work.

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?

This is my first research project. I researched finding optimal parameters to improve the heuristic ant colony algorithm and implement the algorithm. I used such libraries as numpy, matplolib and I also needed to get data for their analysis.

https://github.com/AiGaf1/Study-ACO-for-TSP

7.Did you work with VCS? Which one?

I worked with Git.

8.Did you work with Github Actions?

I am familiar with this but not much.

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

I'm only a little familiar with Docker

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

ed25519 is a digital signature scheme.

An important practical advantage of Ed25519 over ECDSA: The latter family of algorithms completely breaks when used for signatures together with a broken random number generator and Ed25519 has the advantage of being able to use the same key for signing and for key agreement.

11. Do you have any experience in data mining?

I took courses on coursera where mining date was discussed.