

Отчет по лабораторной 2

Результаты для Python:

В процессе выполнения лабораторной работы было протестировано несколько моделей:

- Salesforce/codet5p-220m
- Salesforce/codet5-large-ntp-py
- Salesforce/codet5-large
- Salesforce/codet5-base
- Salesforce/codet5-small
- Salesforce/codet5p-2b
- Salesforce/codet5p-220m-py
- microsoft/codebert-base
- microsoft/unixcoder-base

Проблема заключалась в том, что ни одна из них не давала приемлемый результат. В итоге пришлось реализовать простой фильтр, который убирал невалидные для названия функции символы, а также в случае, если модель вместо токена подставляла несколько слов, брал первое слово. Это значительно улучшило score.

Без комментариев:

Exact Match: 0.142

ROUGE Score: 0.349

С комментариями:

Exact Match: 0.206

ROUGE Score: 0.4585476190476193

ROUGE вырос на 31%, EM вырос на 47%

Некоторые худшие случаи:

ROUGUE1	Reference	Predict
0.0	get_vid_from_url	extract_video_id
0.0	makeMimi	md5
0.0	sprint	format_text
0.0	ckplayer_get_info_by_xml	def

Видно, что модель может галлюцинировать и выдавать ключевые слова языка, но в основном выдает нормальные названия, просто они отличаются от тех, которые были придуманы программистом.

Тело для makeMimi/md5

```
"""From
http://cdn37.atwikiimg.com/sitescript/pub/dksitescript/FC2.site.js
Also com.hps.util.fc2.FC2EncrptUtil.makeMimiLocal
L110"""
strSeed = "gGddgPfeaf_gzyr"
```

```
prehash = upid + "_" + strSeed
return md5(prehash.encode('utf-8')).hexdigest()
```

Видно, что название md5 весьма удачно и отражает суть функции.

Также был проведен аналогичный эксперимент с javascript, но он не увенчался успехом, так как модель выдавала результаты в snake case, а в javascript принято называть функции в camel case.

Результаты для javascript:

Без комментариев:

Exact Match: 0.007

ROUGE Score: 0.01

С комментариями:

Exact Match: 0.008

ROUGE Score: 0.009

Результаты плохие.

Пример работы:

Reference	Predict
xios	createInstance
is_array_buffer	isArrayBufferView
backoff	retryStrategy
main	writePackageManifest