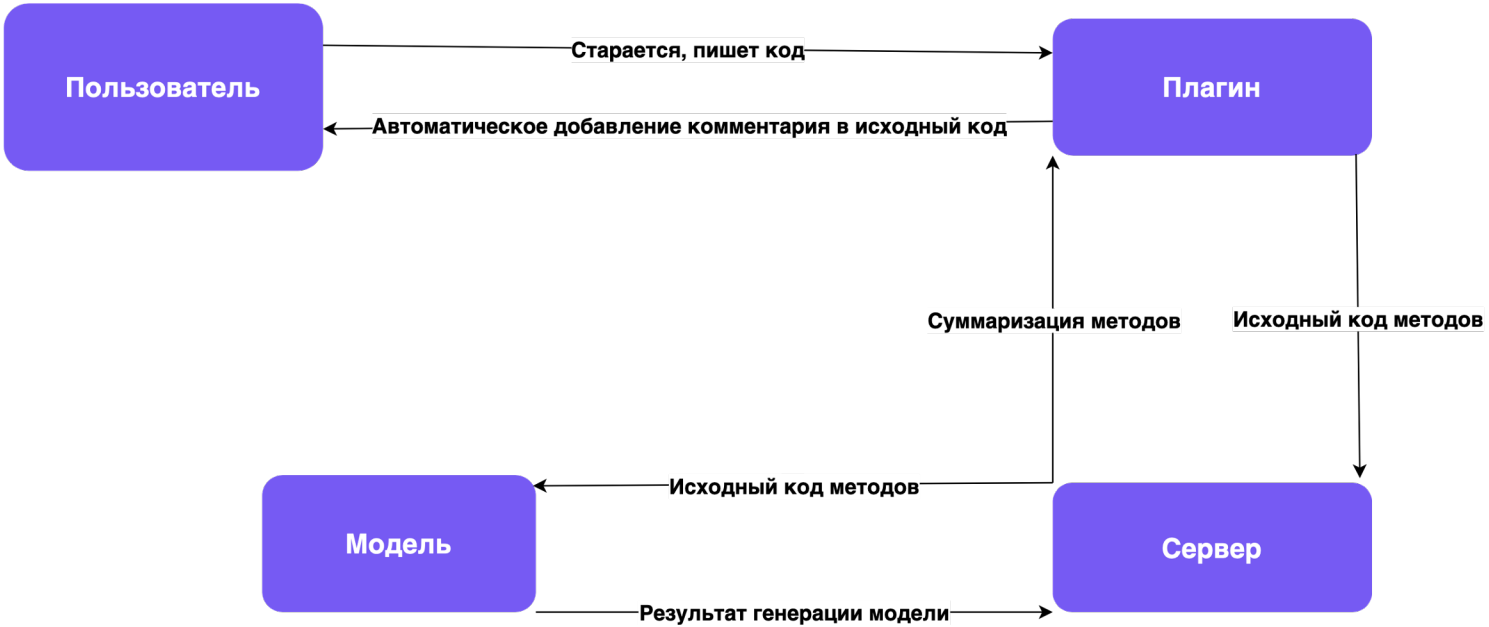
Генерация документации к исходному коду с помощью nlp

Студентка: Лавриченко О. А.
Научный руководитель: Шпильман А.А.
НИУ ВШЭ - Санкт-Петербург

Задача проекта

Цель данного проекта реализовать программу для автоматической генерации документации к методам...

Архитектура проекта



Стек технологий

	Плагин	Сервер	Модель
Функциональность	• Взаимодействие с пользователем • Парсер исходного кода Java • Автоматическое добавление описания методов	• Связь плагина с удаленным местоположением модели • Автоматическое добавление обработанных методов в локальный api ElasticSearch • Создание и отправление запросов по поиску методов	• Предпроцессинг данных • Кодирование последовательностей • Генерация суммаризации исходного кода метода
Стек технологий	Gradle	Flask, docker	PyTorch, Tokenizers

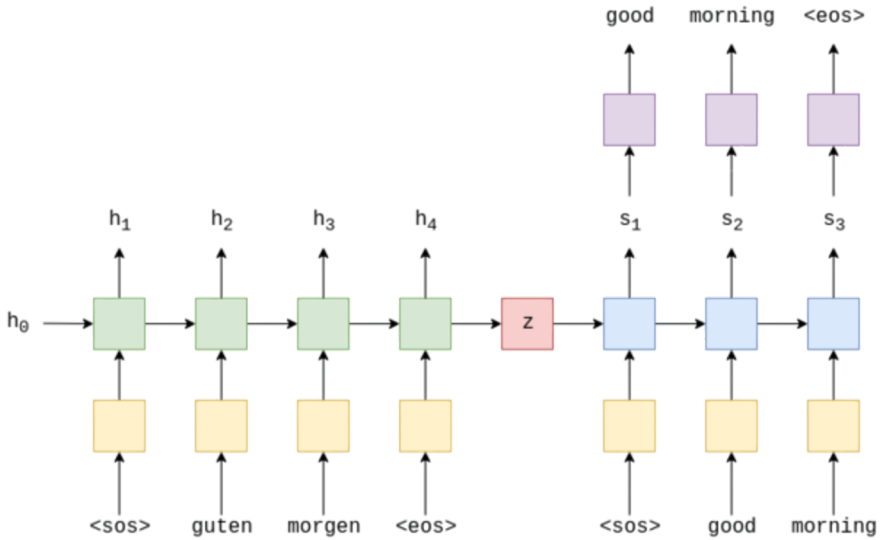
Пайплайн обработки данных

function_id	function	comment
9245436	' public void close() throws IOException {\n input.close();\n }\n'	' /** By default, closes the input Reader. */\n'

```
['[CLS]',
 'public',
 'void',
 'close',
 '(',
 ')',
 'throws',
 'ioexception',
 '{',
 'input',
 '.',
 'close',
 '(',
 ')',
 ';',
 '}',
 '[SEP]']
```

```
[2,
 1480,
 1511,
 2209,
 12,
 13,
 1602,
 1962,
 69,
 2000,
 18,
 2209,
 12,
 13,
 31,
 71,
 3]
```

Модель Seq2Seq



- Embedding layer
- Encoder (LSTM/GRU/RNN/Bi-RNN)
- Final hidden state
- Decoder (LSTM/GRU/RNN/Bi-RNN)
- Linear layer

```
INPUT_DIM = 20000
OUTPUT_DIM = 20000
ENC_EMB_DIM = 256
DEC_EMB_DIM = 256
HID_DIM = 512
N_LAYERS = 2
```

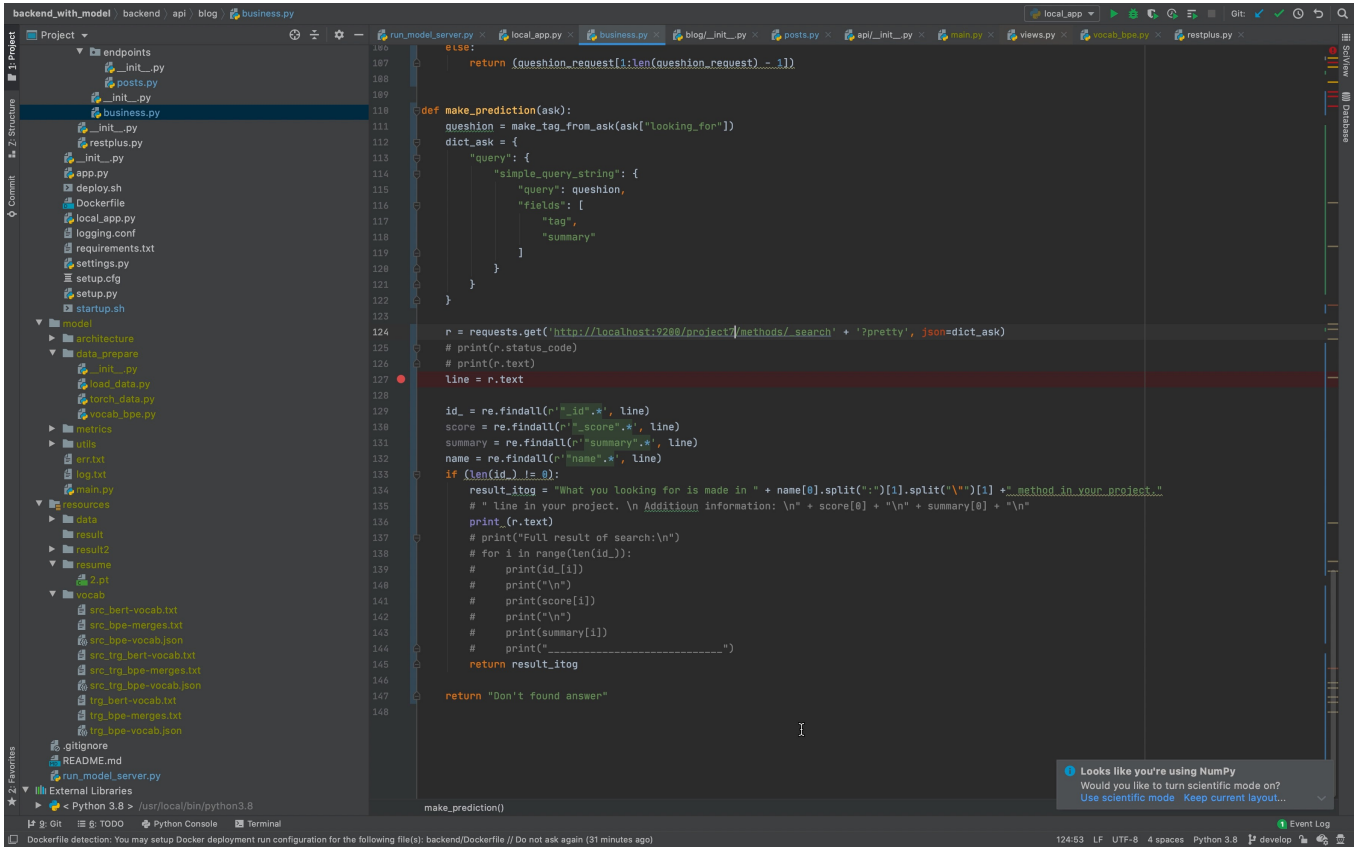
```
Trainable parameters:
27856416
```

Пример работы

```
/**
 * Compares the equality of the object object
 * @param: Object o;
 */
@Override
public boolean equals(Object o) {
  if (this == o) return true;
  if (o == null || getClass() != o.getClass()) return false;

  ProjectDictionary that = (ProjectDictionary)o;

  if (activeName != null ? !activeName.equals(that.activeName) : that.activeName != null) return false;
  if (dictionaries != null ? !dictionaries.equals(that.dictionaries) : that.dictionaries != null) return false;

  return true;
}

/**
 * Returns a hash code value for this object.
 */
@Override
public int hashCode() {
  int result = activeName != null ? activeName.hashCode() : 0;
  result = 31 * result + (dictionaries != null ? dictionaries.hashCode() : 0);
  return result;
}


/**
 * Returns a string representation of the object.
 */
@NonNls
@Override
public String toString() { return "ProjectDictionary{" + "activeName='" + activeName + '\'' + ", dictionaries="
}
```

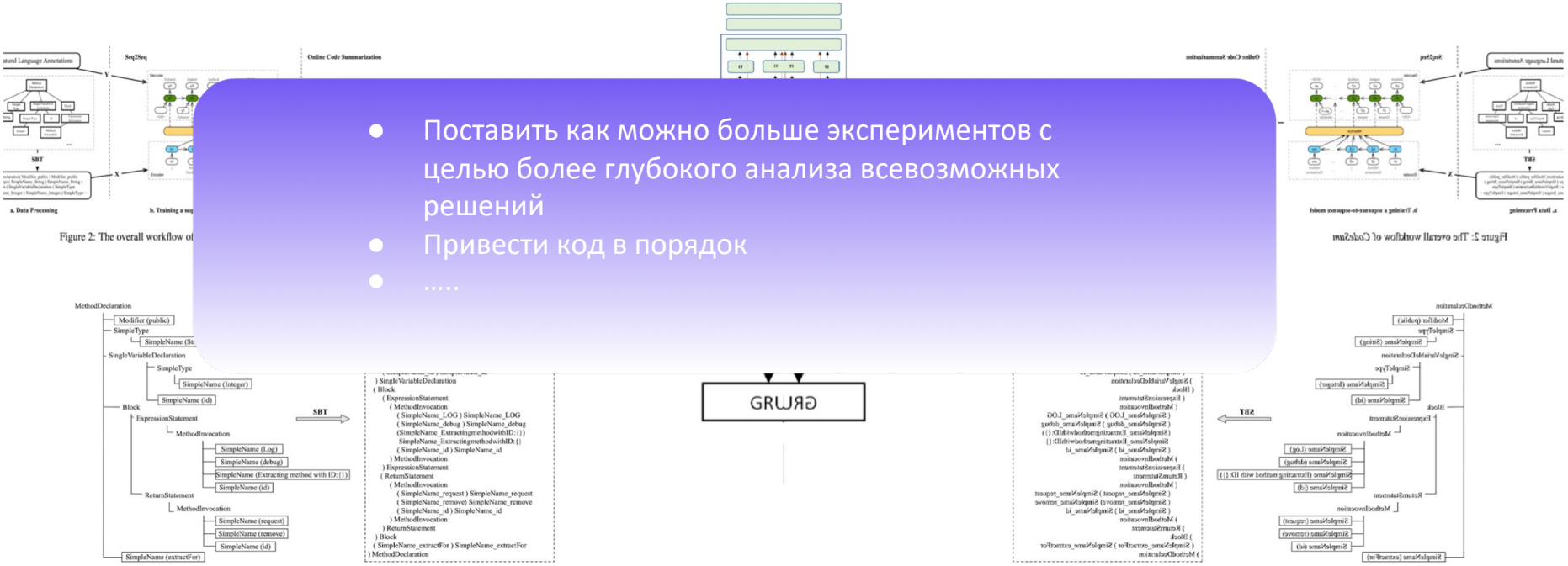
```
/**
 * Process the tree of the the given element.
 * @param: PsiElement element;
 *     HashSet<String> seenNames;
 */
protected void process(@NotNull final PsiElement element, @NotNull final HashSet<String> seenNames) {
  final int endOffset = element.getTextRange().getEndOffset();

  // collect leafs  (spell checker inspection works with leafs)
  final List<PsiElement> leafs = new ArrayList<>();
  if (element.getChildren().length == 0) {
    // if no children - it is a leaf!
    leafs.add(element);
  }
  else {
    // else collect leafs under given element
    PsiElement currentLeaf = PsiTreeUtil.firstChild(element);
    while (currentLeaf != null && currentLeaf.getTextRange().getEndOffset() <= endOffset) {
      leafs.add(currentLeaf);
      currentLeaf = PsiTreeUtil.nextLeaf(currentLeaf);
    }
  }

  for (PsiElement leaf : leafs) {
    processLeafsNames(leaf, seenNames);
  }
}
```

Демо


backend_with_model
backend
api
blog
business.py
Project
endpoints
__init__.py
posts.py
__init__.py
business.py
__init__.py
restplus.py
__init__.py
app.py
deploy.sh
Dockerfile
local_app.py
logging.conf
requirements.txt
settings.py
setup.cfg
setup.py
startup.sh
model
architecture
data_prepare
__init__.py
load_data.py
torch_data.py
vocab_bpe.py
metrics
utils
err.txt
log.txt
main.py
resources
data
result
result2
resume
2.pt
vocab
src_bert-vocab.txt
src_bpe-merges.txt
src_bpe-vocab.json
src_trg_bert-vocab.txt
src_trg_bpe-merges.txt
src_trg_bpe-vocab.json
trg_bert-vocab.txt
trg_bpe-merges.txt
trg_bpe-vocab.json
.gitignore
README.md
run_model_server.py
External Libraries
< Python 3.8 > /usr/local/bin/python3.8
run_model_server.py
local_app.py
business.py
blog/__init__.py
posts.py
api/__init__.py
main.py
views.py
vocab_bpe.py
restplus.py
else:
return (queshion_request[1:len(queshion_request) - 1])
def make_prediction(ask):
queshion = make_tag_from_ask(ask["looking_for"])
dict_ask = {
"query": {
"simple_query_string": {
"query": queshion,
"fields": [
"tag",
"summary"
]
}
}
}
r = requests.get('http://localhost:9200/project7/methods/_search' + '?pretty', json=dict_ask)
# print(r.status_code)
# print(r.text)
line = r.text
id_ = re.findall(r'"_id".*', line)
score = re.findall(r'"_score".*', line)
summary = re.findall(r'"summary".*', line)
name = re.findall(r'"name".*', line)
if (len(id_) != 0):
result_itog = "What you looking for is made in " + name[0].split(":")[1].split("\"")[1] +" method in your project."
# " line in your project. \n Additioun information: \n" + score[0] + "\n" + summary[0] + "\n"
print_(r.text)
# print("Full result of search:\n")
# for i in range(len(id_)):
# print(id_[i])
# print("\n")
# print(score[i])
# print("\n")
# print(summary[i])
# print("_____")
return result_itog
return "Don't found answer"
Looks like you're using NumPy
Would you like to turn scientific mode on?
Use scientific mode
Keep current layout...
make_prediction()
9: Git
6: TODO
Python Console
Terminal
Event Log
Dockerfile detection: You may setup Docker deployment run configuration for the following file(s): backend/Dockerfile // Do not ask again (31 minutes ago)
124:53
LF
UTF-8
4 spaces
Python 3.8
develop

Будущий план

- Поставить как можно больше экспериментов с целью более глубокого анализа всевозможных решений
- Привести код в порядок
- .....



Литература

1. *Jacob Devlin Ming-Wei Chang Kenton Lee Kristina Toutanova* “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding” [https://arxiv.org/pdf/1810.04805.pdf]
2. *Vivek Gupta* “DeepSumm - Deep Code Summaries using Neural Transformer Architecture” [https://arxiv.org/pdf/2004.00998.pdf]
3. *Ilya Sutskever, Oriol Vinyals* “Sequence to Sequence Learning with Neural Networks” [https://arxiv.org/pdf/1409.3215.pdf]
4. *Harshil Shah, David Barber* “Generative Neural Machine Translation” [https://arxiv.org/pdf/1806.05138.pdf]
5. “CodeSum: Translate Program Language to Natural Language∗” [https://arxiv.org/pdf/1708.01837v1.pdf] AST
6. “The Relevance of Software Documentation, Tools and Technologies: A Survey” [https://dl.acm.org/doi/pdf/10.1145/585058.585065]
7. “Learning Source Phrase Representations for Neural Machine Translation” [https://arxiv.org/pdf/2006.14405.pdf]
8. “Neural Machine Translation for Multilingual Grapheme-to-Phoneme Conversion” (https://arxiv.org/pdf/2006.14194.pdf)
9. “Sequence to Sequence Learning with Neural Networks” (https://arxiv.org/pdf/1409.3215.pdf)
10. “Effective Approaches to Attention-based Neural Machine Translation” [https://arxiv.org/pdf/1508.04025.pdf]
11. CODE-NN [https://www.aclweb.org/anthology/P16-1195.pdf]