

## Постановка задачи

Требуется разработать retrieval-based чат-бота, который отвечает на вопросы реплика определенного персонажа сериала.

**Github проекта:** [https://github.com/YurezSml/MIPT\\_NLPGener\\_HW1](https://github.com/YurezSml/MIPT_NLPGener_HW1)

## Описание и анализ данных

В качестве рассматриваемого сериала был выбран сериал «Друзья»

Источник данных: <https://www.kaggle.com/datasets/gopinath15/friends-netflix-script-data>

Датасет состоит из 5 столбцов

Text – реплика персонажа и технический текст

Speaker – персонаж, говорящий реплику

Episode – номер и название эпизода

Season – номер сезона

Show – название шоу

Всего датасет содержит 69974 записи. Как будет видно далее, из анализа, не все они являются репликами персонажей, часть можно отнести к техническим записям, связанным с рекламными вставками, окончанием серии и т.д. Несколько записей исходного датасета приведены на рисунке 1.

```
Ввод [123]: 1 df = pd.read_csv('Friends.csv')
            2 df
```

Out[123]:

	Text	Speaker	Episode	Season	Show
0	Originally written by Marta Kauffman and David...	NaN	Episode-01-The One Where Monica Gets a New Roo...	Season-01	Friends
1	Transcribed by guineapig.	NaN	Episode-01-The One Where Monica Gets a New Roo...	Season-01	Friends
2	CENTRAL PERK. (ALL PRESENT EXCEPT RACHEL AND ...	SCENE 1	Episode-01-The One Where Monica Gets a New Roo...	Season-01	Friends
3	There's nothing to tell! He's just some guy I...	MONICA	Episode-01-The One Where Monica Gets a New Roo...	Season-01	Friends
4	C'mon, you're going out with the guy! There's...	JOEY	Episode-01-The One Where Monica Gets a New Roo...	Season-01	Friends
...	...	...	...	...	...
69969	Then I'm happy too. (They're still hugging - ...	Ross	Episode-15-The One Where Estelle Dies	Season-10	Friends
69970	COMMERCIAL BREAK	NaN	Episode-15-The One Where Estelle Dies	Season-10	Friends
69971	Estelle's memorial service. Joey is giving a ...	[Scene	Episode-15-The One Where Estelle Dies	Season-10	Friends
69972	Thank you all for coming. We're here today to...	Joey	Episode-15-The One Where Estelle Dies	Season-10	Friends
69973	THE END	NaN	Episode-15-The One Where Estelle Dies	Season-10	Friends

69974 rows × 5 columns

Рисунок 1 – Исходный датасет, состояние до обработки

В ходе анализа и первичной обработки датасета все имена авторов реплик были приведены к одному виду. В датасете обнаружены пропущенные значения для поля с авторами реплик (6284), в ходе анализа было принято решение использовать часть из них в качестве возможных разделителей при разбиении датасета на несвязанные диалоги. Остальные записи с пропущенными авторами реплик были удалены, как разрывающие контекст

внутри диалога. Кроме того, в качестве разделителей между диалогами были приняты специальные отметки в поле ‘Text’.

Результаты анализа и первичной обработки представлены в файле [https://github.com/YurezSml/MIPT\\_NLPGener\\_HW1/blob/main/Analysys.ipynb](https://github.com/YurezSml/MIPT_NLPGener_HW1/blob/main/Analysys.ipynb)

## Подготовка данных

В результате первичной обработки был подготовлен датасет, состоящий из двух полей:

Text – реплика персонажа и технический текст

Speaker – персонаж, говорящий реплику

Независимые или слабо зависимые по контексту диалоги отделены с помощью разделителя 'break' в поле ‘Speaker’ (Рисунок 2). Нельзя говорить о полной независимости диалогов друг от друга т.к. возможно продолжение общего контекста как через несколько диалогов, так и в разных сериях. Чтобы устранить это и сделать максимально связный контекстом датасет с репликами требуется значительное количество времени, ручного труда и погружение в сюжет.

	Text	Speaker
0	central perk.	break
1	there's nothing to tell! he's just some guy i ...	monica
2	c'mon, you're going out with the guy! there's ...	joeey
3	so does he have a hump? a hump and a hairpiece?	chandler
4	wait, does he eat chalk?	phoebe
...	...	...
63484	yeah, yeah, oh!	ross
63485	oh! oh, i'm so happy.	rachel
63486	then i'm happy too.	ross
63487	estelle's memorial service. joeey is giving a s...	break
63488	thank you all for coming. we're here today to ...	joeey
63489 rows × 2 columns		

Рисунок 2 – Датасет после первичной обработки

Полученный датасет позволяет выбрать реплики любого из персонажей в качестве объекта для обучения. В своей работе я выбрал Чендлера (chandler), поскольку его реплики и шутки являются одними из самых ярких в сериале и выделяются на фоне остальных. Реплики персонажей преобразовывались в датасет для обучения следующим образом: в качестве ответа записывается реплика выбранного персонажа. В поле с вопросом записываются

предшествующие реплики других персонажей, но не более заданного количества реплик REPL\_DEPTH, определяющего глубину контекста для реплики. Если между выбранным персонажем и другими происходит диалог, то предыдущие пары вопрос-ответ записываются в поле с контекстом, но не более заданного количества пар CONT\_DEPTH, определяющего глубину контекста диалога. На рисунке 3 представлено распределение количества реплик в зависимости от глубины контекста диалога. Дополнительное исследование показало, что максимальный из диалогов имеет 84 предшествующих пары вопрос-ответ.

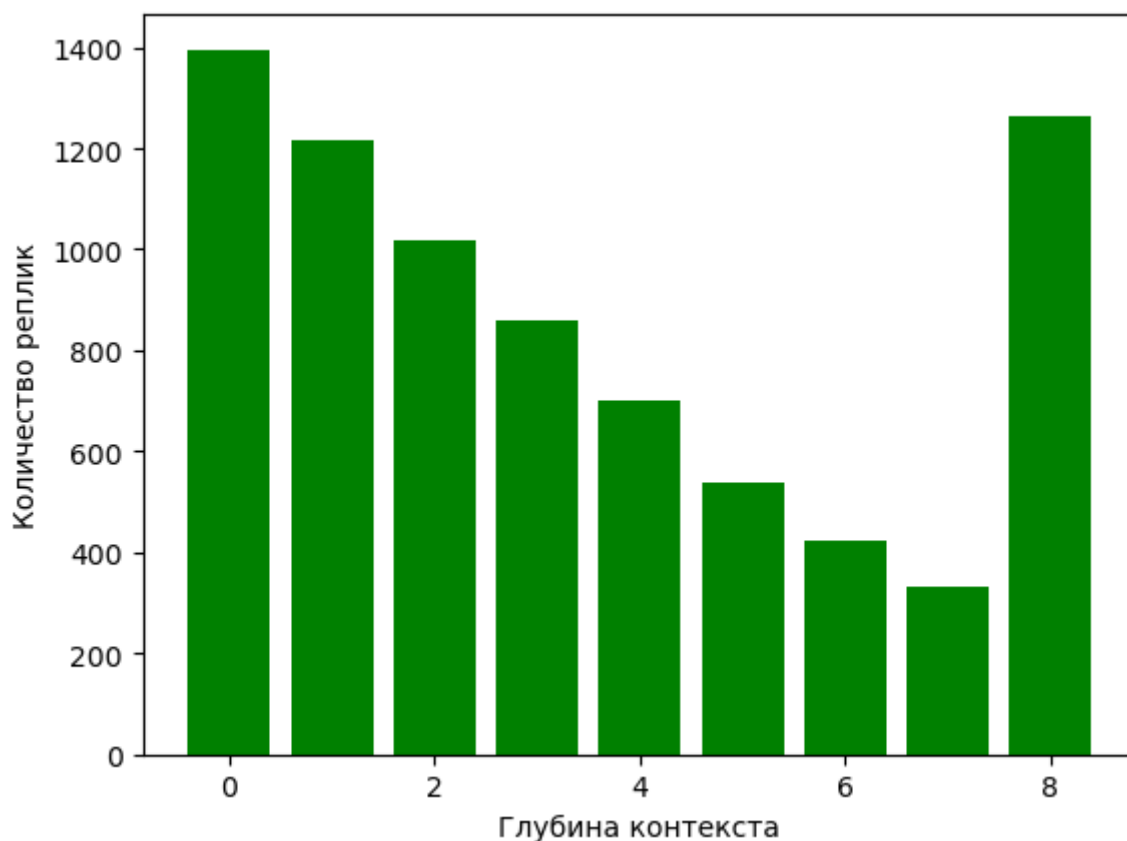


Рисунок 3 – Количество реплик от глубины контекста диалогов в итоговом датасете

## Обучение модели

В качестве модели для дообучения я выбрал distilbert. Также пробовал дообучать bert, но скорость обучения на доступном мне железе сильно ниже. Модель дополнялась линейным слоем, в токенизатор добавлялся спецтокен [U\_token]. При подготовке датасета для обучения с помощью [U\_token] отделялись реплики, соответствующие вопросам и ответам в контексте. Для определения максимальной длины последовательности в токенайзере анализируем распределение вопросов и ответов в зависимости от их длины (Рисунок 4). По результатам анализа максимальная длина была установлена равной 512.

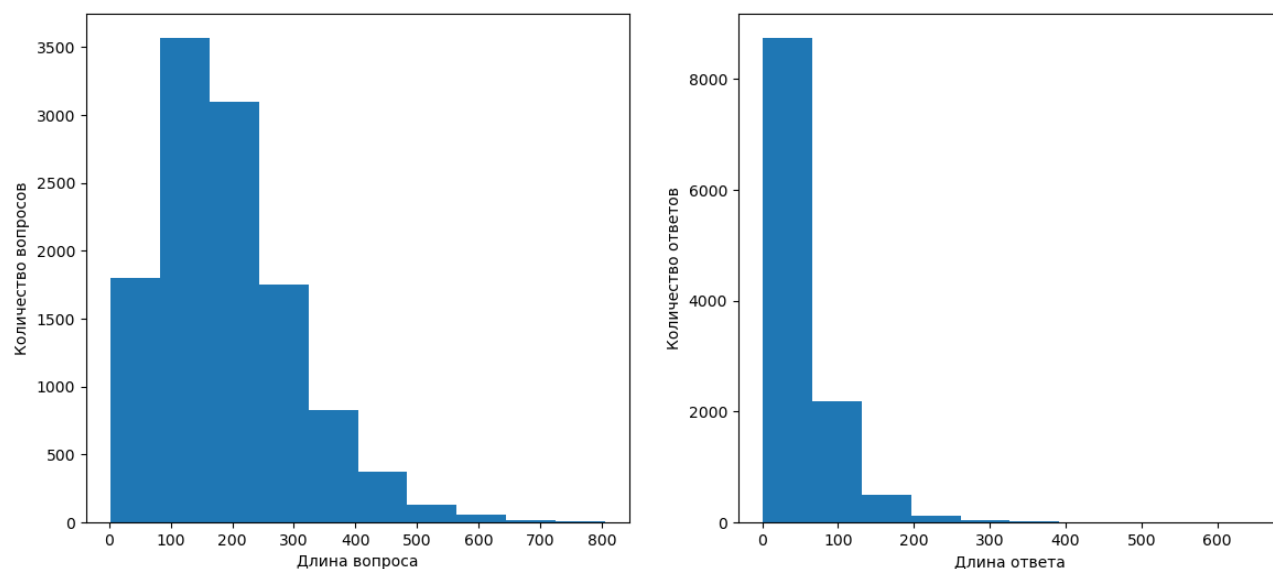


Рисунок 4 – Количество вопросов/ответов от их длины

Модель обучалась одну эпоху т.к. более длительное обучение (анализировалось вплоть до 10 эпох) не оказало существенного влияния на результат. В ходе обучения модели возникло предположение, что она возможно попадает в локальный минимум и можно попытаться преодолеть его, увеличив шаг обучения. Было проанализировано влияние шага и шедулера на результаты обучения, результаты представлены на рисунках 5 – 8. Можно сделать вывод, что наиболее стабильный результат получается при обучении с линейным шедулером и шагом, равным  $1E-4$ . Для всех вариантов обучение проводилось несколько раз, для анализа воспроизводимости результатов, для косинусного шедулера было получено, что результат обучения может довольно сильно изменяться от одного старта к другому. Примеры этого представлены на рисунке 8.

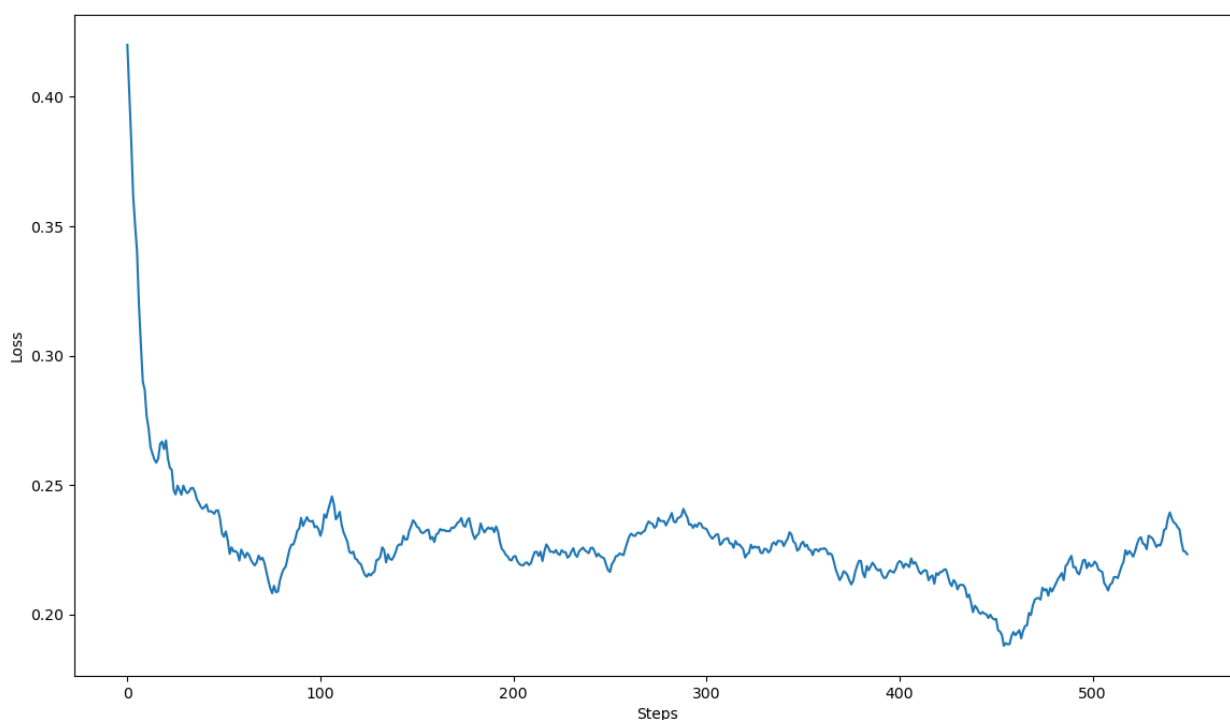


Рисунок 5 – График обучения модели, линейный шедулер, learning rate = 3E-5

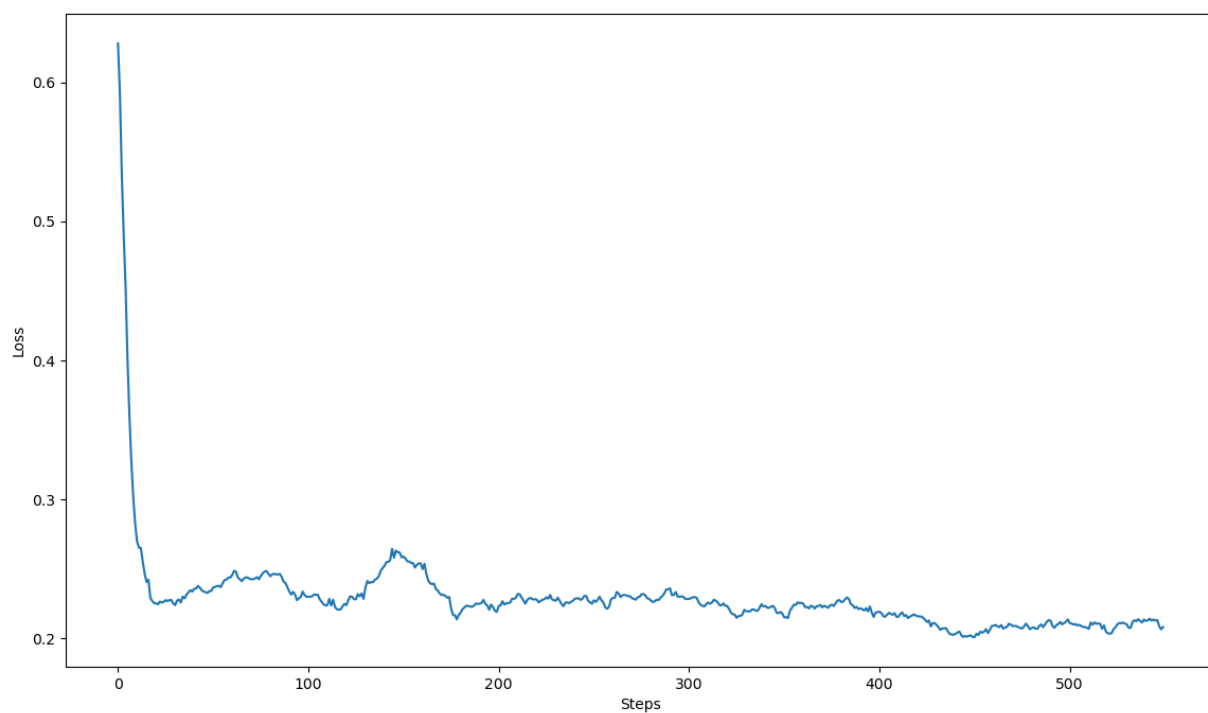


Рисунок 6 – График обучения модели, линейный шедулер, learning rate = 1E-4

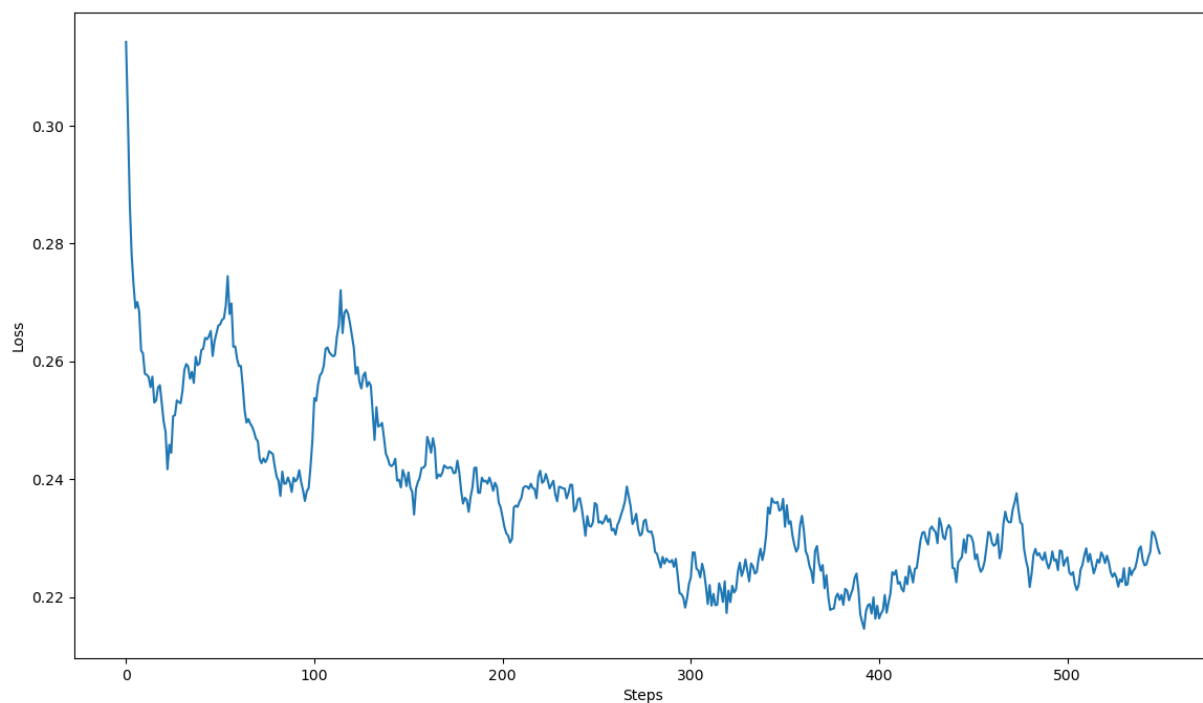


Рисунок 7 – График обучения модели, линейный шедулер, learning rate = 5E-4

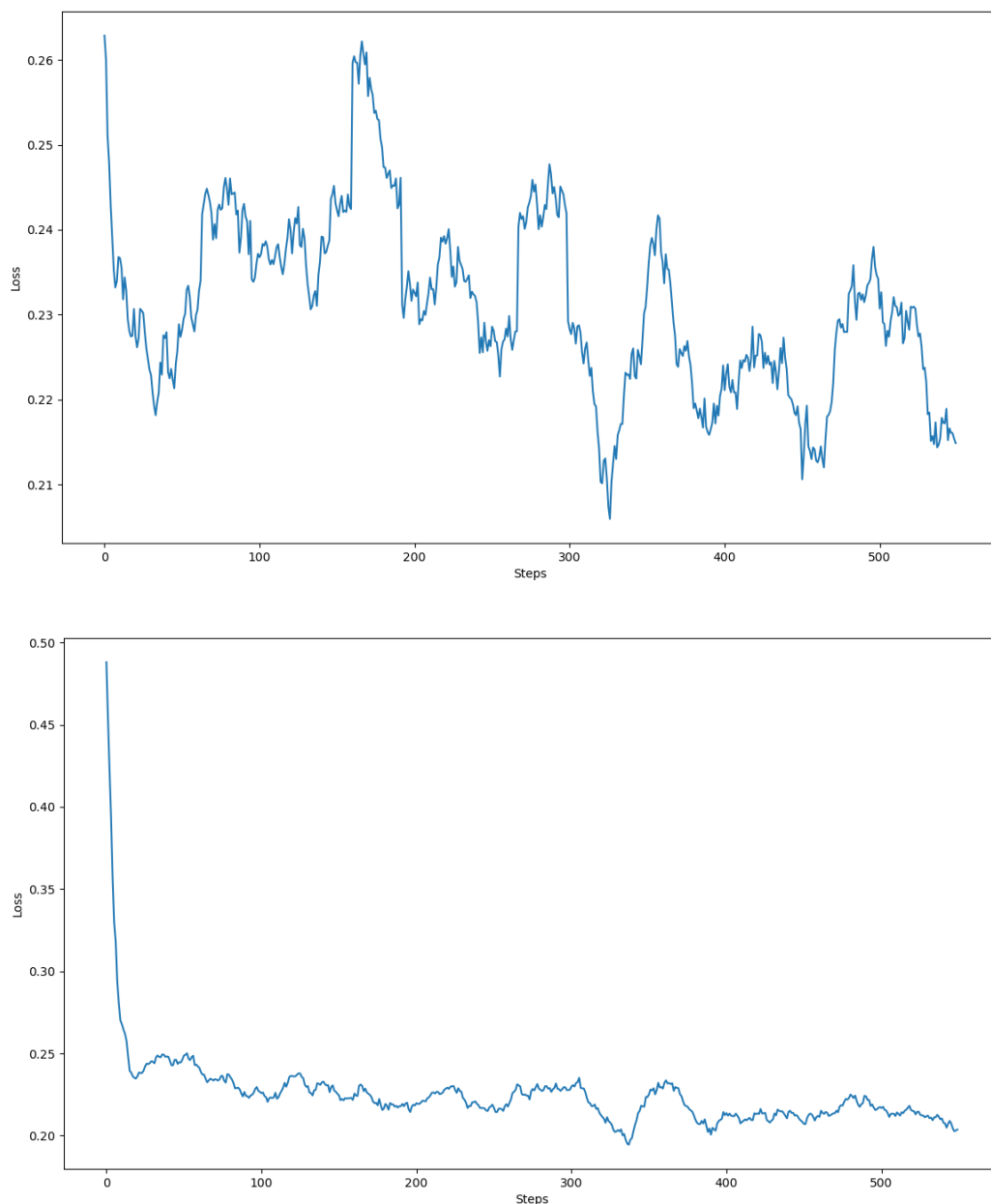


Рисунок 8 – Графики обучения модели, косинусный шедулер, learning rate = 1E-4

## Инференс и валидация модели

Инференс модели реализован в виде вебсервиса с четырьмя методами:

1. ping – проверка работоспособности сервера
2. request – возвращает 1 наиболее релевантный ответ
3. retrieve – возвращает 5 наиболее релевантных ответов
4. dialog – разработан для демонстрации возможности бота вести диалог, принимает 3 вопроса и последовательно выдает на них ответы, с учетом контекста предыдущих пар вопрос-ответ

Кроме того, реализованы два возможных способа сбора корпуса реплик для выбора наиболее релевантных.

1. В качестве корпуса подаются все реплики персонажа (медленная реализация)
2. В качестве корпуса выбирается и подается установленное число реплик (по умолчанию равное 500), наиболее близких к вопросу. Близость к вопросу определяется с помощью KNN. (быстрая реализация)

Для валидации результатов выбраны следующие тесты:

1. Простое приветствие ('hey!'), медленная реализация  
Проверка ответа бота на простой вопрос и оценка времени работы медленного алгоритма. Результат представлен на рисунке 9. Результат выглядит разумно, на приветствие следует простое приветствие.

```
request:      "'hey!'"
response:     "hey!"
response_code: "200"
time:         "22.00682306289673"
```

Рисунок 9 – Результат работы чат-бота на тестовой задаче

2. Простое приветствие ('hey!'), быстрая реализация  
Проверка ответа бота на простой вопрос и оценка времени работы быстрого алгоритма. Результат представлен на рисунке 10. Результат в чем-то соответствует сути персонажа, на приветствие следует ответ в виде шутки и нелепой ситуации. Видно, что скорость выполнения увеличилась в 3 раза.

```
request:      "'hey!'"
response:     "out of my league. i could get a brian. [brian enters behind him] if i wanted to get a brian, i could get a brian. [sees him] hey, brian."
response_code: "200"
time:         "7.793093681335449"
```

Рисунок 10 – Результат работы чат-бота на тестовой задаче

3. Частый вопрос другого персонажа из диалогов в сериале ('oh my god!', Дженис или Джо), медленная реализация  
Проверка ответа бота на встречающийся в исходном датасете вопрос. Результат представлен на рисунке 11. По смыслу похоже на возможный ответ на такое восклицание.

```
request:      "'oh my god!'"
response:     "y'know what? i can't believe this! do you know what you did? my girlfriend is out there thinking things over! you made my girlfriend think!!!"
response_code: "200"
time:         "22.498421907424927"
```

Рисунок 11 – Результат работы чат-бота на тестовой задаче

4. Частый вопрос других персонажей из диалогов в сериале ('what are you doing?'), быстрая реализация

Проверка ответа бота на встречающийся в исходном датасете вопрос. Результат представлен на рисунке 12. В целом большинство реплик подходят по смыслу в качестве ответов на вопрос.

```
request:      "'what are you doing?'"
response:
  0:      "oh! some guy. some guy. 'hey jill, i saw you with some guy last night. yes, he was some guy.'"
  1:      "uh, if i were omnipotent for a day, i'd.. make myself omnipotent forever."
  2:      "well, i have an appointment to see dr. robert pillman, career counselor a-gogo. [pause] i added the 'a-gogo'."
  3:      "wait a minute, wait a minute, i see where this is going, you're gonna ask him to new year's, aren't you. you're gonna break the pact. she's gonna break the pact."
  4:      "i'm smoking. i'm smoking, i'm smoking."
  5:      "i don't care, i don't care! game's over! i'm weak! i've gotta smoke! i've gotta have the smoke!"
  6:      "hey, you guys in the living room all know what you want to do. you know, you have goals. you have dreams. i don't have a dream."
  7:      "well, what? what? what is it? that she left you? that she likes women? that she left you for another woman that likes women?"
  8:      "no! uh, i d'know! the point is, if you were gonna set me up with someone, i'd like to think you'd set me up with someone like him."
  9:      "out of my league. i could get a brian. [brian enters behind him] if i wanted to get a brian, i could get a brian. [sees him] hey, brian."
response_code: "200"
time:      "16.586353302001953"
```

Рисунок 12 – Результат работы чат-бота на тестовой задаче

#### 5. Диалог из трех реплик, медленная реализация

Проверка работы бота на имитации реального диалога. Результат представлен на рисунке 13. Диалог получился не очень связным, возможно из-за простоты и однотипности вопросов.

```
requests:
  0:      "'hello!'"
  1:      "'how are you?'"
  2:      "'what're you doing this evening?'"
response:
  0:      "wait a minute, is she going for spring vacation or is she going for spring break?"
  1:      "no-no-no-no-no!"
  2:      "well, i-i thought i did but, i-i guess i did not!"
response_code: "200"
time:      "126.21325469017029"
```

Рисунок 13 – Результат работы чат-бота на тестовой задаче

## Выводы

На основе retrieval подхода разработан чат-бот. Дообученная на датасете из реплик сериала «Друзья» модель distilbert оценивает релевантность реплик, далее ранжирующий механизм отбирает из них наиболее релевантные. Реализовано решение на базе метода к-ближайших соседей (KNN), позволяющее ускорить работу чат-бота за счет предварительного отбора реплик персонажа, наиболее близких к задаваемому вопросу. Чат-бот реализован в виде веб-интерфейса с четырьмя методами, позволяющими как задавать боту отдельные вопросы, так и направлять сразу несколько вопросов для поддержания диалога. Работоспособность бота продемонстрирована на тестовых примерах. Качество работы оценивалось исходя из адекватности ответов заданному вопросу.

## Инструкция



Инференс модели реализован в виде вебсервиса с четырьмя методами:

1. ping  
не получает на вход параметры. возвращает "status": 'working!' в случае успешного подключения
2. request  
получает на вход 2 параметра через GET: query и type. Параметр type может принимать значения 'fast' или 'slow'. В случае успешного срабатывания возвращает  
"response\_code": "200"  
"request": query  
"response": ответ\_модели  
"time": время выполнения запроса
3. retrieve – возвращает заданное количество релевантных ответов  
получает на вход 3 параметра через GET: query, type и num\_answers. Параметр type может принимать значения 'fast' или 'slow'. В случае успешного срабатывания возвращает  
"response\_code": "200"  
"request": query  
"response": num\_answers\_релевантных\_ответов\_модели  
"time": время выполнения запроса
4. dialog – разработан для демонстрации возможности бота вести диалог, принимает 3 вопроса и последовательно выдает на них ответы, с учетом контекста предыдущих пар вопрос-ответ  
получает на вход 4 параметра через GET: query\_1, query\_2, query\_3 и type. Параметр type может принимать значения 'fast' или 'slow'. В случае успешного срабатывания возвращает  
"response\_code": "200"  
"request": список\_вопросов  
"response": список\_ответов  
"time": время выполнения запроса