

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

Факультет информатики, математики и компьютерных наук

**Программа подготовки бакалавров по направлению
01.03.02 Прикладная математика и информатика**

Добряев Иван Александрович

КУРСОВАЯ РАБОТА

Применение предобученной модели GPT-3 в задаче обработки текстов

Научный руководитель

Старший преподаватель

Пономаренко А.А.

Консультант

Симагин Д. А.

ООО «Яндекс.Технологии»

Нижний Новгород, 2021

Введение

В наше время, количество текстов, которые нужно обработать человеком, превосходит его физические возможности, поэтому в данный момент стремительно развивается такая отрасль как NLP (Netural Language Preprocessing), смысл которой в автоматизации препроцессинга естественного языка и обработки больших текстов. Например: задача перевода текста на другой язык.

Сегодня абсолютное большинство людей сталкивалось с программами собеседниками в том или ином виде: это может быть голосовой ассистент или бот поддержки банка. Как правило, они выполняют конкретную цель: помочь пользователю в поставленном вопросе. Но 'общаясь' с ними, люди понимают, что это не человек, поскольку такие боты имеют определенную задачу и не способны поддержать обычный диалог.

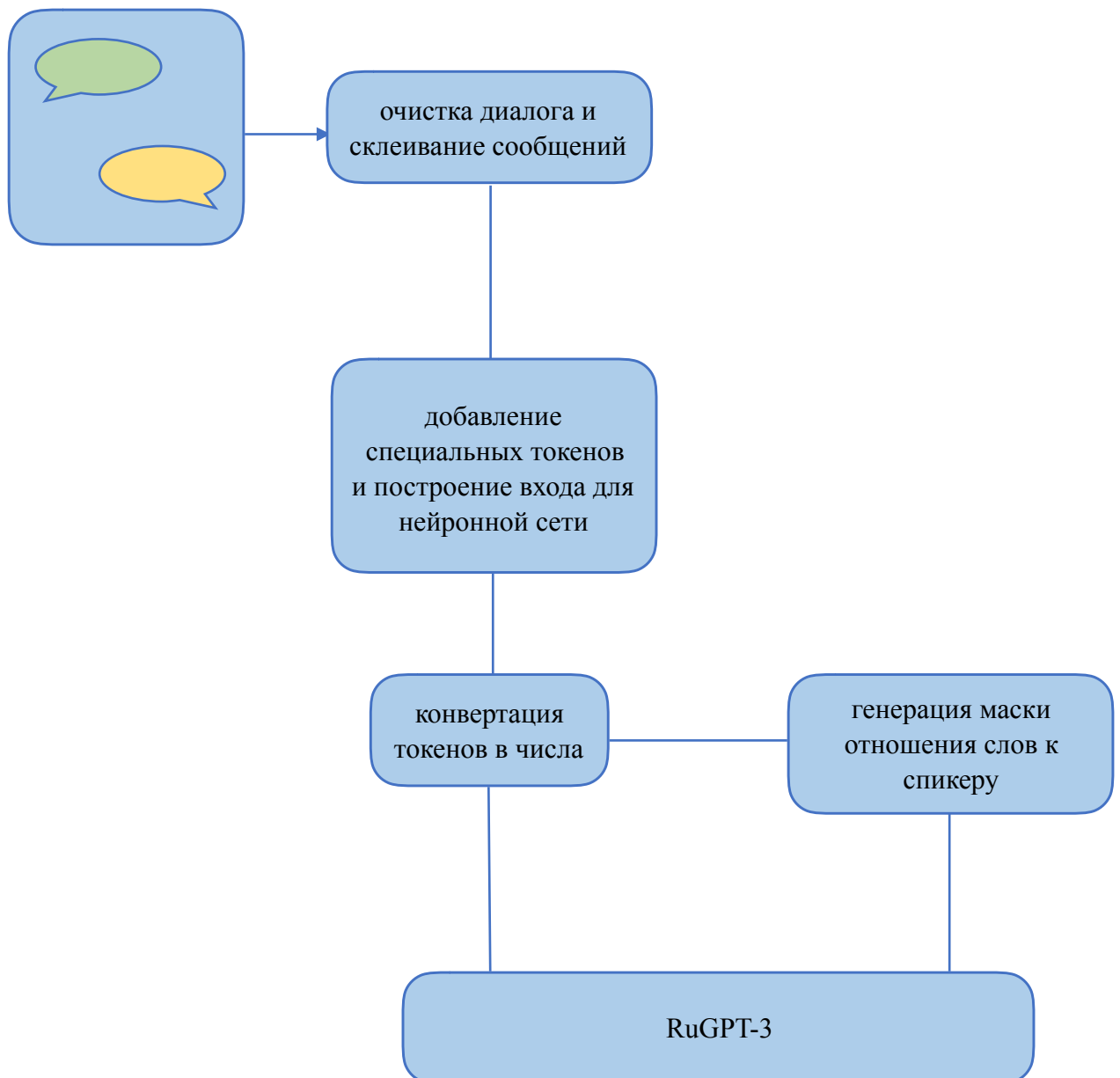
В скором будущем, когда роботы на улице будут обыденностью, чтобы быть более человекоподобными, они должны будут уметь разговаривать как обычные люди, поэтому важной задачей является создание диалоговых систем, которые будут почти что неотличимы от человека.

Также такие программы собеседники важны для людей которым не хватает общения. Ярким примером является главный герой фильма "Бегущий по лезвию 2049", жена которого являлась программой, заменяющая ему живого человека.

Итогом исследовательской работы будет являться программа написанная на языке Python с использованием крупнейшей нейросетью-трансформером `gpt-3` обученной на русском языке, с возможностью поддержать диалог с человеком.

Подготовка данных

Для поставленной задачи был выбран датасет [\[1\]](#) "*Toloka Persona Chat Rus*", собранный лабораторией нейронных сетей и глубокого обучения МФТИ. Набор данных содержит профили воображаемых личностей с описаниями и диалогами между участниками.



*схема предобработки данных перед
поступлением в нейросеть*

- *очистка диалога и склеивание сообщений:*

Поскольку данные в датасете хранятся в HTML тэгах, пришлось очистить их.

- *добавление специальных токенов и построение входа для нейронной сети:*

Для того, чтобы нейросеть могла выделять для себя информацию необходимо добавить специальные разделяющие токены:

- **<bos>** - биография человека, который отвечает и начало входных данных.
- **<speaker1>**, **<speaker2>** - разделения между репликами
- **<fos>** - начало генерации ответа
- **<eos>** - конец диалога

- *конвертация токенов в числа:*

С помощью специального предустановленного токенайзера переводим каждое слово в токен, используя таблицу соответствия, где каждому токenu присвоен уникальный эмбединг - представление токена виде набора чисел.

- *генерация маски отношения слов к спикеру:*

Для улучшения работы ruGPT-3, генерируется вектор состоящий из нулей и единиц, показывающий к какому спикеру относится то или иное слово.

Пример:

Спикер 1: Меня зовут Олег, мне 32 года. Работаю полицейским

Спикер 2: Я Женя, учусь в 9 классе

Диалог:

1: Привет!

1: Как тебя зовут?

2: Привет, я Женя, а тебя как?

1: Меня Олег, приятно познакомиться!

Вход модели:

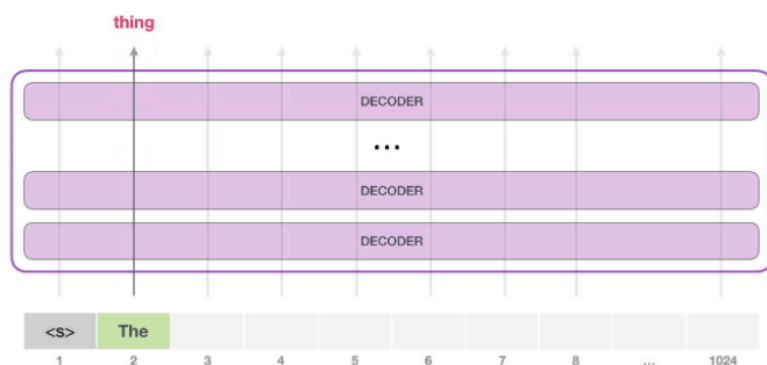
<bos>Меня зовут Олег, мне 32 года. Работаю полицейским<speaker1> Привет! Как тебя зовут?
<speaker2> Привет, я Женя, а тебя как?<fos>Меня Олег, приятно познакомиться!<eos>

<speaker1>...<speaker1><speaker2>...<speaker2><speaker1>...<speaker1>

Ru-GPT-3 medium

Ru-GPT-3 medium [2] основана на модели от OpenAi GPT-2, отличие которой заключается в том, что ее обучили на русском языке. Чтобы лучше понять как она работает, рассмотрим ее архитектуру и идеи, которые в ней используются.

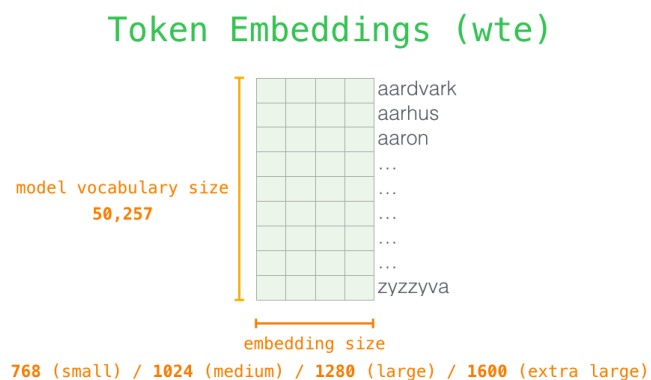
Вся нейронная сеть из блоков Decoder, значения которых передаются из одного в другой. Выходом модели является вектор вероятностей для каждого слова, что оно является продолжением предложения, поданного на вход.



Рассмотрим GPT-2 более детально:

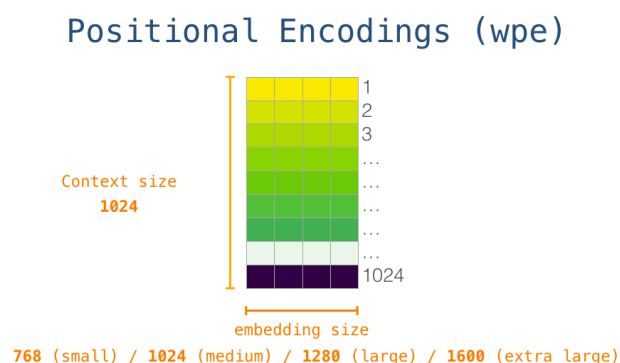
- Кодирование входа:

Для начала модель сопоставляет эмбеддинг для каждого слова в своей матрице эмбеддингов, так как нейросеть не может работать со словами. Это может быть как и слово целиком, так и его часть.



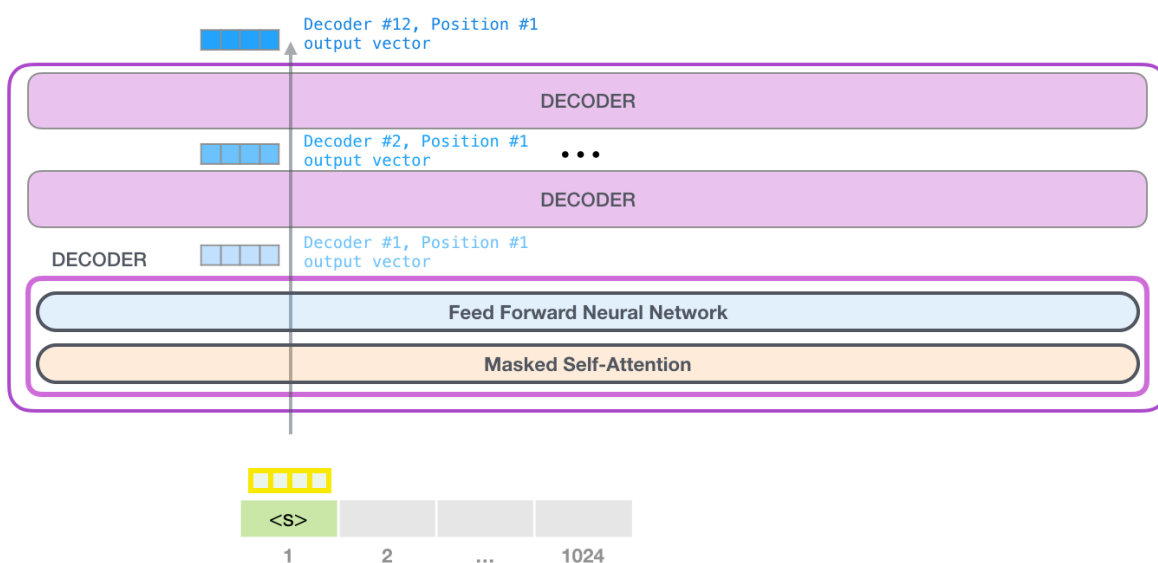
- Позиционное кодирование:

Перед тем чем передать вектор токенов в первый блок модели, надо добавить позиционное кодирование – маска, указывающая на порядок слов в предложении. Частью обученной модели также является матрица, содержащая позиционные кодирующие векторы для каждой из 1024 позиций во входе.



- Блок Decoder:

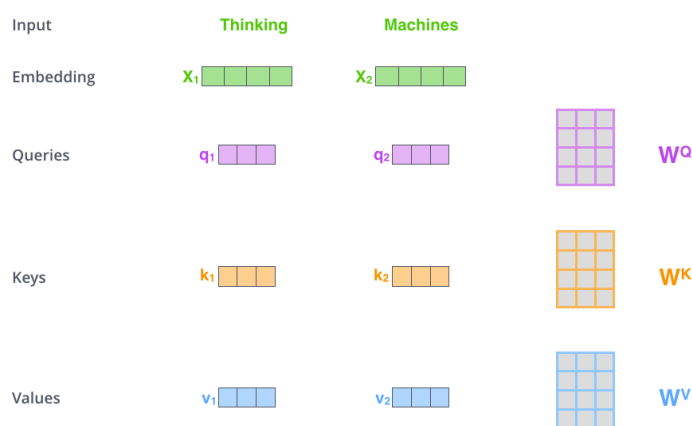
Блок Decoder состоит из двух подблоков Masked Self-Attention и Feed Forward Neural Network, данные в которые подаются последовательно.



- Self-Attention:

Основной идеей трансформеров является механизм Self-Attention. Благодаря которому и достигается такой невероятный результат.

Первый этап в подсчете внутреннего внимания — это создание трех векторов из эмбединга каждого слова : вектор запроса (Query vector), вектор ключа (Key vector) и вектор значения (Value vector). Эти векторы задаются с



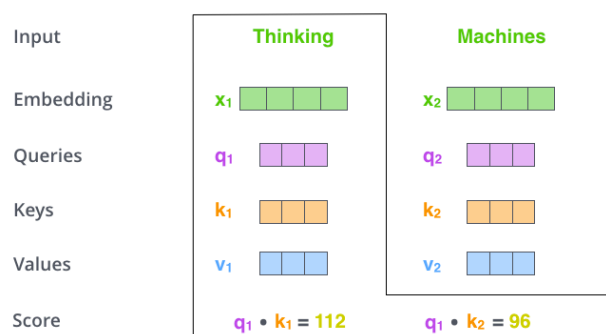
помощью перемножения эмбединга на три матрицы, которые подбираются во время обучения.

Обратите внимание, что размерность полученных векторов меньше, чем векторы эмбедингов. Их размерность составляет 64, в то время как эмбединги и входящие/выходящие векторы энкодера имеют размерность 1024 (в нашем случае).

Второй этап внутреннего внимания – формирование коэффициента (score). Предположим, в данный

момент подсчитывается внутреннее внимание для слова – «Thinking». Нужно оценить каждое другое слово предложения по отношению к данному. Коэффициент помогает

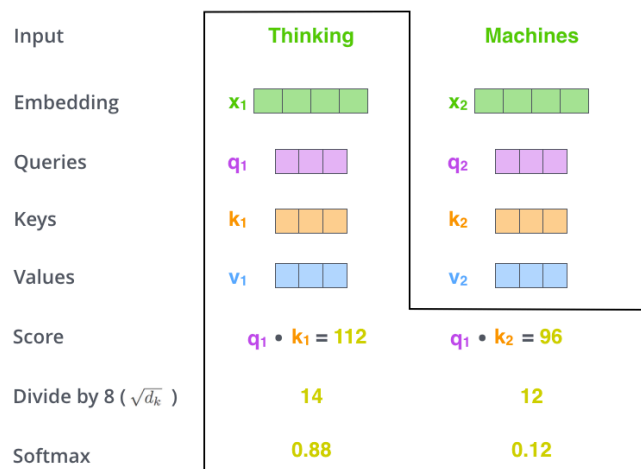
определить насколько стоит сфокусироваться на других токенах в предложении во время кодирования слова в конкретной позиции.



Коэффициент рассчитывается скалярным произведением вектора запроса на вектор ключа соответствующего слова.

Третий этап – поделить полученные коэффициенты на квадратный корень размерности векторов ключа. Данное значение обеспечивает более стабильные градиенты. После чего рассчитать функцию

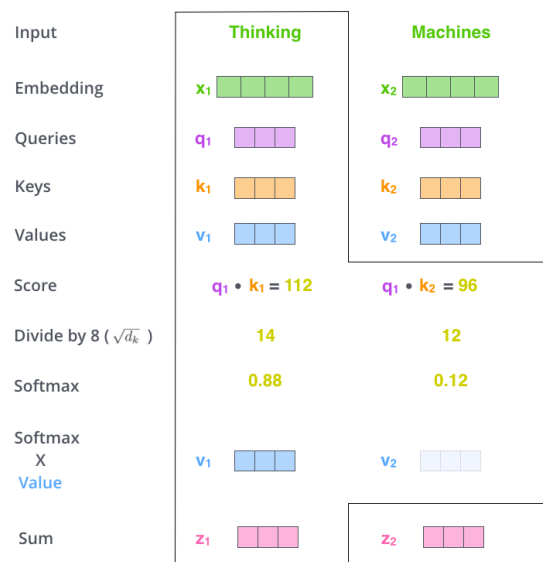
$$\text{softmax}(x_i) = \frac{\exp(x_i)}{\sum_{j=1}^n \exp(x_j)}.$$



Данная функция нормализует коэффициенты так, чтобы они были положительными и в сумме давали 1.

Четвертый этап — домножить каждый вектор значения на значение софтмакс. Основная идея заключается в том, чтобы держать без изменений значения слов, на которых фокусируется модель, и отвести на второй план нерелевантные слова.

Пятый этап — сложить полученный взвешенный вектор и вектор значения. Это и будет представлять собой выход слоя внутреннего внимания в данной позиции (для первого слова).

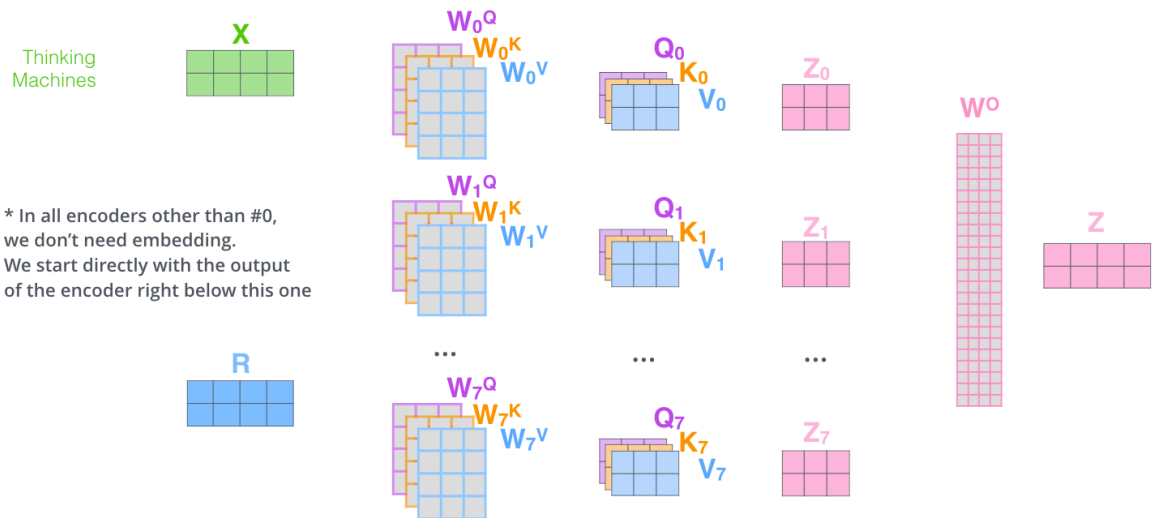


- Multi-head attention:

Данная технология улучшает производительность внутреннего слоя внимания за счет следующих пунктов:

1. Повышается способность модели фокусироваться на разных позициях.
2. Слой внимания снабжается множеством «подпространств представлений» (representation subspaces).

- 1) This is our input sentence*
- 2) We embed each word*
- 3) Split into 8 heads. We multiply X or R with weight matrices
- 4) Calculate attention using the resulting $Q/K/V$ matrices
- 5) Concatenate the resulting Z matrices, then multiply with weight matrix W^O to produce the output of the layer



Основное отличие self-attention от masked self-attention, заключается в том, что при подсчета score не учитываются слова, которые стоят впереди.

Scores (before softmax)				Masked Scores (before softmax)			
0.11	0.00	0.81	0.79	0.11	-inf	-inf	-inf
0.19	0.50	0.30	0.48	0.19	0.50	-inf	-inf
0.53	0.98	0.95	0.14	0.53	0.98	0.95	-inf
0.81	0.86	0.38	0.90	0.81	0.86	0.38	0.90

Apply Attention Mask

- Feed Forward Neural Network:

Полносвязная нейронная сеть – это сеть, где скрытые слои обрабатывают входной токен после того, как внутреннее внимание учло контекст в его представлении. Сеть состоит из двух скрытых слоев. Первый слой в 4 раза больше размера входа gpt-3 (поскольку medium gpt-3 имеет размерность 1024, эта нейронная сеть будет иметь $768 * 4 = 4096$ нейронов)

Второй слой преобразует результат первого блока в размерность модели (1024). Результатом работы полносвязанной нейронной сети будет выход блока Трансформера для данного токена.

Обучение

Обучение происходило в среде Google Colab, с оптимайзером Adam и функцией целевой ошибки Cross Entropy Loss $L_{CE} = - \sum_{i=1}^C y_i \log \hat{y}_i$, где

- \hat{y}_i - i -ая вероятность из softmax.
- y_i - бинарное значение, принимающее 1 только при истинном токене.

В датасете находится примерно 10000 диалогов. Одна эпоха занимает около 30 минут. ruGPT-3 обучалась примерно 10 эпох. Минимальная ошибка составляла: 2.03 .

Примеры диалогов

Изначально, концепция нейронной сети была другой: Человек вводил информацию о себе, а нейросеть старалась завести диалог интересный пользователю. Вот пример одного из первых диалогов:

Расскажите о себе:

слушаю рок, программист, иногда рисую

Начните диалог

-Привет! Как тебя зовут?

-Меня Андрей). Чем увлекаешься в свободное время?. Я люблю читать книги, очень часто рисовать картины). Люблю путешествовать и общаться. А ты чем

-Приятно познакомиться, Я Иван. люблю погулять иногда, только погода не очень(

-😂 а я мечта побывать во всем мире) но мне бы тоже надо было лучше всего ходить по городам, так что моя профессия будет связана с дизайном

-в каких странах уже был?

-? Ты откуда? У меня есть машина или квартира.. Да у нас дом на севере города, поэтому там все получится)))))) В каком городе жив

-я из Москвы!

Очень приятно, Ваня это же здорово!!! Это круто))) ну ладно спасибо за разговор Спасибо тебе удачи! Удачи!). Было пообщаемся) До свидания

Комментарий: в данном диалоге, можно заметить, что нейросеть запомнила мое имя и применила его при прощании.

После, пришла идея дать личность самой нейросети, благодаря добавление описания характера и рода деятельности специальным токеном в начало диалога. Получилось достаточно хорошо.

Описание бота:

Меня Зовут Анна. мне 20 лет, студентка. люблю петь. Учусь на дизайнера

привет! Как дела?

- Привет Хорошо А у тебя? Чем занимаешься?).

Все хорошо, сижу дома. Как тебя зовут?

- Меня зовут Анна, мне 21 год, я студентка 😊. А тебя как?.

Я Ваня. Приятно познакомиться. Аня, ты любишь животных?

- Люблю животных, у меня есть попугай 🦜. Нет, не люблю животных.

Какой породы твой попугай?

- Среднеазиатская овчарка, очень умная и преданная собака, обожаю ее. как ее зовут?

- Анюта, она очень красивая и умная собака. Я очень люблю свою собаку, а ты?

У меня нет собаки(живу в маленькой квартире.

-

Какие фильмы ты смотришь?

-

алло?

- Я люблю смотреть сериалы, особенно про любовь, но больше всего мне нравятся фильмы про путешествия.

А мне больше нравятся детективы.

- Я тоже люблю путешествовать, поэтому часто езжу в разные страны, в том числе и в Китай.

Ого, и как там?

- *Очень интересно и познавательно. В Китае очень много интересных мест, которые я обязательно посещаю.*

Ты была на Великой Китайской стене?

- *Да, это очень интересное место, там очень красиво и красиво.*

А как тебе еда? Говорят там все остро.

- *Я не ем мясо, только рыбу и морепродукты, потому что это вредно для здоровья.*

Это очень здорово! Ладно, мне пора идти спать. Спокойной ночи!

- *Пока, Аня. Спасибо за вечер! 😊..*

Комментарий: в данном диалоге, можно заметить, что нейросеть иногда при ответе смотрит на свои же реплики, думая, что это ответы человека. Также в конце 'Аня' обратилась к человеку, используя свое имя. Это происходит из-за того, что хранятся лишь последние 4 реплики (существует ограничение на кол-во токенов, которые подаются на вход) и описание личности бота.

почти SOTA диалог:

Описание бота:

Меня Зовут Анна. мне 20 лет, студентка. люблю петь. Учусь на дизайнера

Привет!

- Привет! Меня зовут Анна, мне 21 год, я студентка!. А ты кем работаешь?

Я работаю маркетологом, приятно познакомиться

- А я работаю дизайнером, а ты чем занимаешься?. Чем увлекаешься в свободное время?

В свободное время читаю книги и смотрю фильмы? Какой твой любимый фильм?

- Да, люблю смотреть фильмы и читать книги, но больше всего я люблю слушать музыку, особенно рок.

Какая песня твоя любимая?

- Я очень люблю рок, поэтому часто слушаю рок группы, в том числе и группу Ария.

В каких странах ты была?

- Я была в Италии, Испании, Франции, Германии, Австрии, Чехии, Польше, России.

Есть ли у тебя друзья?

- Нет, у меня нет друзей, потому что я не люблю общаться с людьми.

Комментарий: в данном диалоге можно заметить, что ruGPT-3 осознает принадлежность группы "Ария" к рок музыке и знает названия стран.

Попытка построить такой же диалог с Алисой:

-Привет! Как тебя зовут? Кем работаешь?

-Офисным планктоном

-А я работаю дизайнером, чем занимаешься в свободное время?

-На диване лежу, музыку слушаю

какая твоя любимая песня?

вонтс ту лив форевер. Шоу маст гоу он.

Айм ин лов виз май кар. Виз май кар, Карл.

Конечно, Квин.

-Круто! в каких странах ты была?

-Вам перечислить?

-да, давай

-Какой вы любопытный!

-у тебя есть друзья?

-К вам я отношусь как другу.

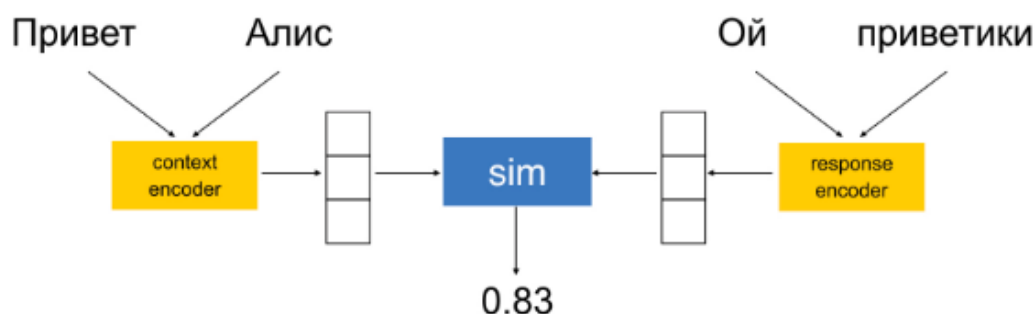
-а кроме меня?

-У меня много друзей и я не считала сколько.

Комментарий: как можно заметить, Алиса отвечает очень кратко, поэтому построить конструктивного диалога с ней не получится.

Другие методы реализации программы собеседника

- Яндекс "Алиса" [\[3\]](#)



Используется DSSM(*Deep Structured Semantic*) [\[4\]](#) - нейронная сеть, в которой имеется два энкодера. Где один кодирует контекст диалога, в то время как другой кодирует возможный ответ. В итоге на выходе получается два embedding-вектора для ответа и для контекста. Сеть обучается таким образом, что метрика косинусного расстояния между полученными векторами было тем больше, чем более релевантен данный ответ в текущем контексте.

Плюсы данной реализации:

- интуитивно понятный подход выбора ответа

Минусы данной реализации:

- Зачастую одинаковый ответ на одну и ту же реплику
- Конечный список ответов

- Apple "Siri", Google "Google Assistant", Amazon "Alexa"

Перечисленные выше компании в своих ассистентах используют технологию, которая называется "Граф знаний" [\[5\]](#) Основная идея которого заключается в переходе из одной вершины (состояния) в другую.

Плюсы данной реализации:

- Можно проследить почему сделан тот или иной выбор

Минусы данной реализации:

- Сложно конструировать графы

Заключение

Можно заметить, что нейронная сеть не является идеальной. В большей степени это связано с достаточно маленьким и однообразным датасетом или с не способностью `gpt-3 medium` генерировать естественную речь. Например, использование более крупной нейросети `gpt-3 large` могло бы улучшить результат . Также пока не существует функции ошибки, которая бы определяла "Правильность" сгенерированного ответа. Поэтому в будущем возможно стоит присмотреться к другим концептуальным идеям генерации естественной речи.

Приложения

исходный код проекта:

<https://github.com/LongaBonga/ChatBot-RUGPT-3>

Список используемой литературы и данных

1. база данных <https://toloka.ai/datasets>
2. The Illustrated GPT-2 (Visualizing Transformer Language Models) // Jay Alammar [Электронный ресурс]. Режим доступа: <https://jalammar.github.io/illustrated-gpt2/> (дата обращения 12.08.2019)
3. Unsupervised Deep Structured Semantic Models for Commonsense Reasoning // Shuohang Wang , Sheng Zhang , Yelong Shen , Xiaodong Liu , Jingjing Liu , Jianfeng Gao , Jing Jiang [Электронный ресурс]. Режим доступа <https://arxiv.org/pdf/1904.01938.pdf> (дата обращения 3.03.2019)
4. Как устроена Алиса // Борис Янгель [Электронный ресурс]. Режим доступа: <https://habr.com/ru/company/yandex/blog/349372/> (дата обращения 18.02.2018)
5. KG-BART: Knowledge Graph-Augmented BART for Generative Commonsense Reasoning // Ye Liu¹ , Yao Wan² , Lifang He³ , Hao Peng⁴ , Philip S. Yu¹
¹Department of Computer Science, University of Illinois at Chicago, Chicago, IL, USA ²School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan, China ³Department of Computer Science and Engineering, Lehigh University, Bethlehem, PA, USA ⁴Beijing Advanced Innovation Center for Big Data and Brain Computing, Beihang University, Beijing, China [Электронный ресурс]. Режим доступа <https://arxiv.org/pdf/2009.12677.pdf> (дата обращения 21.01.2021)