МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

им. Н.Э. Баумана

Факультет «Информатика и системы управления»

Кафедра «Систем обработки информации и управления»

ОТЧЕТ

**Домашнее задание**

по дисциплине«Методы машинного обучения в автоматизированных системах обработки информации и управления»

ИСПОЛНИТЕЛЬ: \_\_Се Цзявэнь\_\_\_\_

ФИО

группа ИУ5И-22М\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

подпись

"\_\_"\_\_\_\_\_\_\_\_\_202\_ г.

ПРЕПОДАВАТЕЛЬ: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

ФИО

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

подпись

"\_24\_"\_\_\_\_\_05\_\_\_\_2024\_ г.

Москва - 2024

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# Задания

Домашнее задание по дисциплине направлено на анализ современных методов машинного обучения и их применение для решения практических задач. Домашнее задание включает три основных этапа:

* выбор задачи;
* теоретический этап;
* практический этап.

# Выбор задачи и теоретический этап

* **"Attention Is All You Need"**

Transformer — это архитектура модели машинного обучения, особенно подходящая для обработки задач обработки естественного языка (NLP). Это модель, основанная на механизме внимания, впервые предложенная Васвани и др. в 2017 году. Ядром модели Transformer является механизм самообслуживания, который может устанавливать ассоциации между различными позициями входной последовательности, чтобы лучше понять контекстную информацию в последовательности. Трансформатор состоит из кодера и декодера. Кодер отвечает за преобразование входной последовательности в скрытое представление, а декодер генерирует целевую последовательность на основе выходных данных кодера. Благодаря своим сильным возможностям распараллеливания и способности обрабатывать зависимости на больших расстояниях Transformer стал предпочтительной модельной архитектурой для многих задач НЛП, таких как машинный перевод, генерация текста и понимание языка.

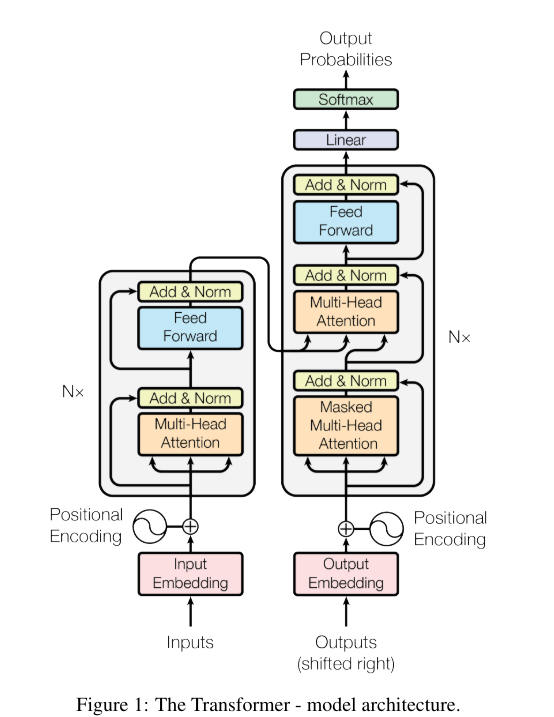
Архитектура модели Transformer состоит из кодеров и декодеров, каждый из которых имеет несколько уровней. В кодере входная последовательность сначала проходит через уровень внедрения, который преобразует каждое слово или токен в соответствующее ему векторное представление. Затем эти векторы дополняются позиционным кодированием, чтобы модель сохраняла информацию о порядке во входной последовательности.

Затем встроенное представление входной последовательности обрабатывается через несколько уровней самообслуживания. Самообслуживание позволяет каждому входному местоположению взаимодействовать с другими местоположениями и назначать веса важности на основе корреляции между ними. Это помогает модели уловить зависимости между частями входной последовательности.

За слоем самообслуживания следует нейронная сеть прямого распространения, которая выполняет нелинейное преобразование представления каждой позиции.

Структура декодера аналогична кодировщику, но также включает в себя дополнительный многоголовочный уровень самообслуживания для обработки информации от кодера и выходной уровень для генерации целевой последовательности.

Каждый уровень во всей архитектуре модели Transformer использует остаточные соединения и нормализацию слоев для улучшения эффекта обучения и распространения градиента.

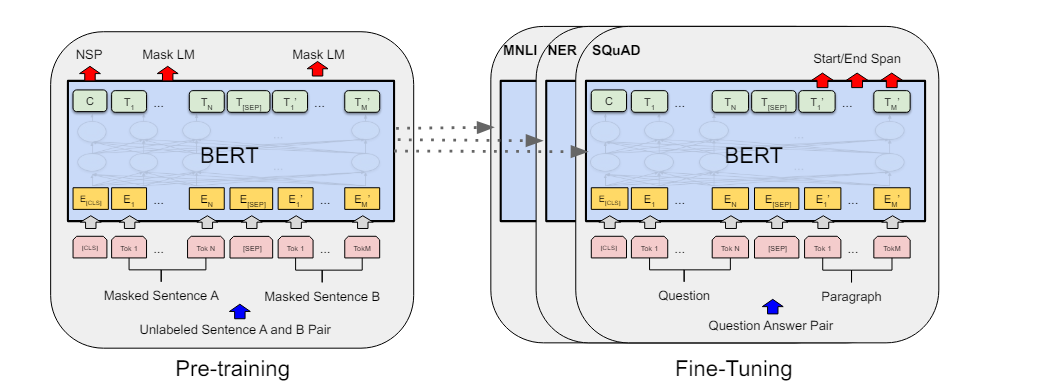
****

* **BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding**

BERT (представления двунаправленного кодировщика от Transformers) — это предварительно обученная модель обработки естественного языка, выпущенная Google в 2018 году. Это модель, основанная на архитектуре Transformer, которая предварительно обучается с использованием крупномасштабных неразмеченных текстовых данных, изучает представления языка и настраивается для различных задач НЛП для достижения впечатляющей производительности.

Основная идея BERT — использовать двунаправленную контекстную информацию для генерации словесных представлений, а не полагаться только на односторонний контекст. Для достижения этой цели BERT использует двунаправленный преобразователь-кодер. Он использует двунаправленный механизм самообслуживания, который позволяет модели одновременно учитывать все остальные позиции каждой позиции во входной последовательности, тем самым лучше улавливая семантическую и грамматическую информацию в предложении.

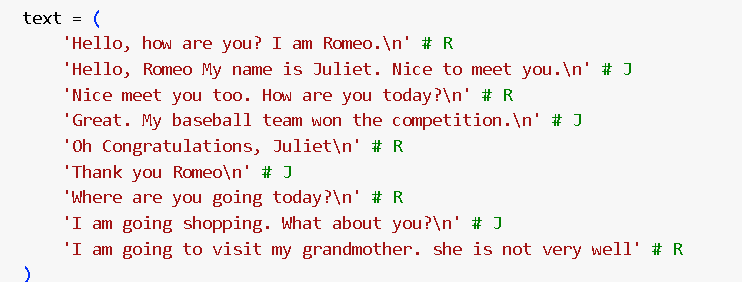
Процесс предварительного обучения BERT разделен на два этапа: модель языка в маске (MLM) и прогнозирование следующего предложения (NSP). На этапе MLM модель случайным образом маскирует некоторые слова во входном предложении и пытается предсказать эти замаскированные слова. На этапе NSP модель принимает пару предложений в качестве входных данных и пытается предсказать, являются ли эти два предложения последовательными. Благодаря этим двум этапам предварительного обучения BERT может изучить богатые языковые представления и продемонстрировать хорошие общие результаты при выполнении различных последующих задач.



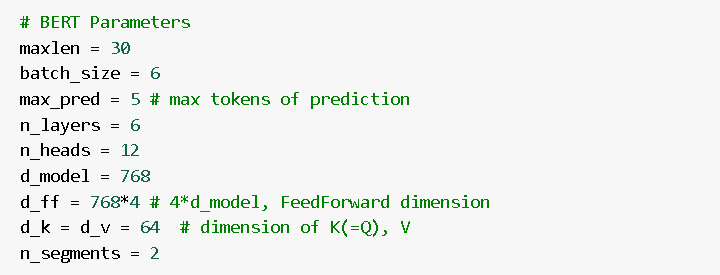
# Практический этап

* Подготовить набор данных

Используя простой набор данных, разговор между двумя людьми был введен вручную.

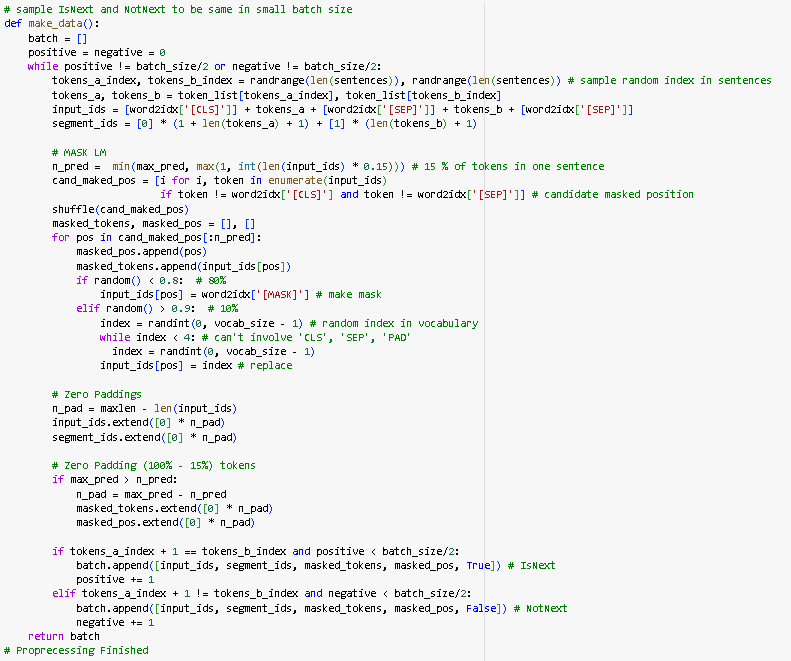


* Параметры модели

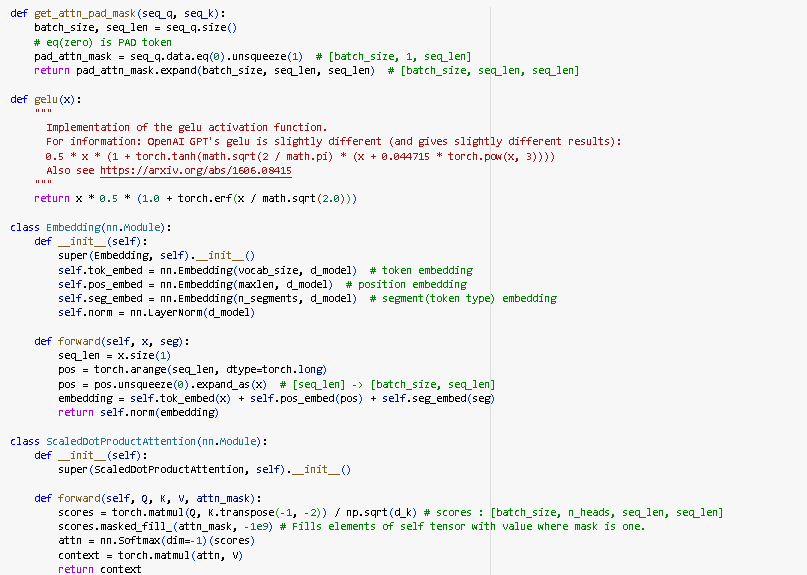


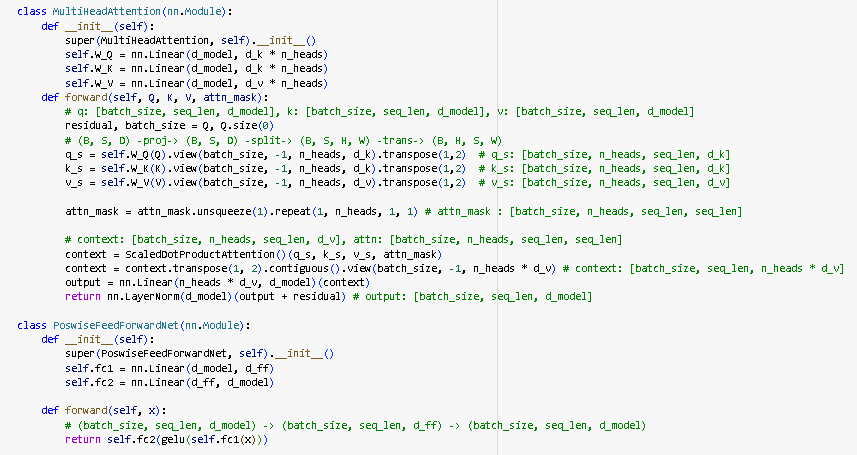
* Предварительная обработка данныха

В части предварительной обработки данных нам необходимо случайным образом покрыть или заменить (далее совместно именуемые «покрытие») 15% токенов в предложении в соответствии с вероятностью, а также нам нужно соединить любые два предложения.



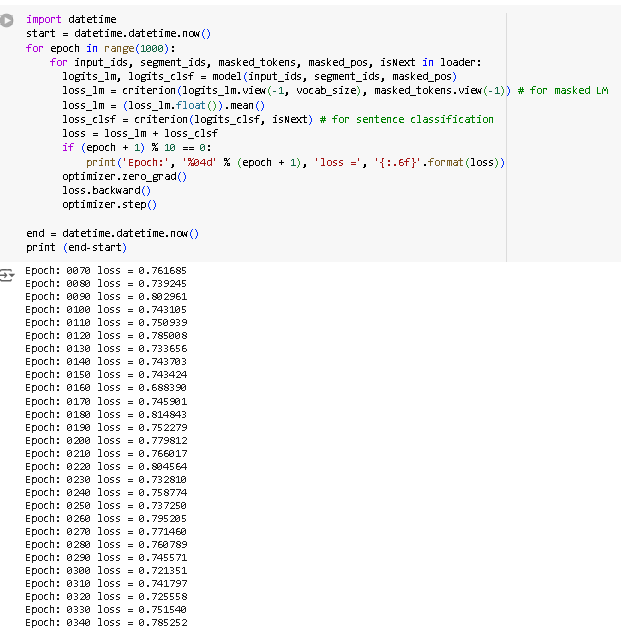
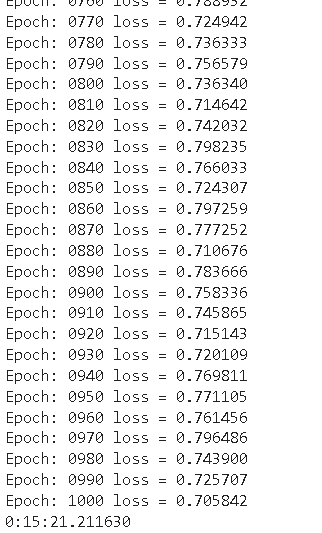
* Построение модели

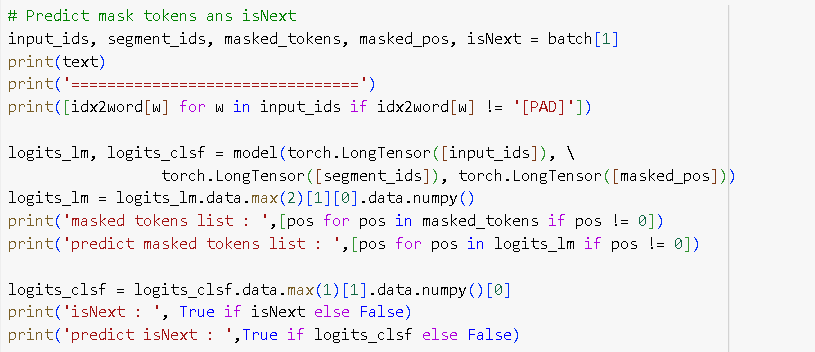


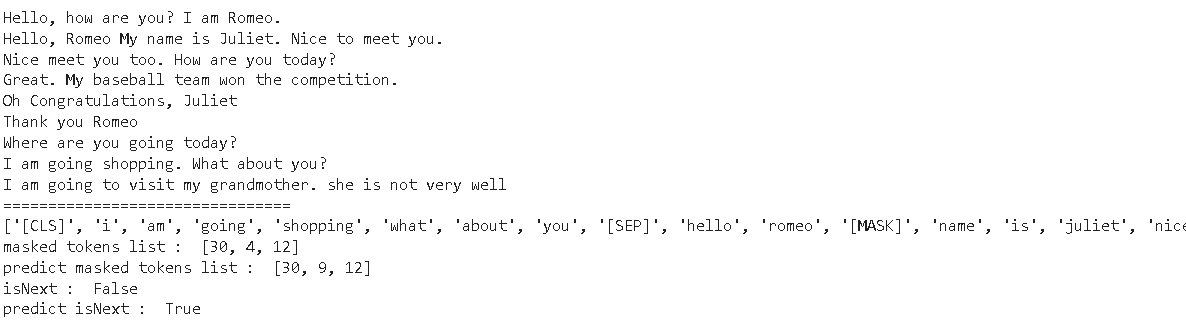




* Обучение и тестирование





# Вывод

В этом задании я решил проанализировать и воспроизвести модели Transformer и BERT и применить их к простому набору данных разговора. На теоретическом этапе я глубоко изучил принципы работы и применения моделей Transformer и BERT, включая их конкретные топологии, математические описания, функциональные алгоритмы и процессы обучения.

На практическом этапе я воспроизводил эксперименты соответствующих работ, использовал исходный код, предоставленный автором, и записывал результаты работы программы.

# ПИСОК ИСТОЧНИКОВ

1. @article{DBLP:journals/corr/abs-1810-04805,author = {Jacob Devlin and Ming{-}Wei Chang and Kenton Lee and Kristina Toutanova}, title = {{BERT:} Pre-training of Deep Bidirectional Transformers for Language Understanding}, journal = {CoRR}, volume = {abs/1810.04805}, year = {2018}, url ={http://arxiv.org/abs/1810.04805}, archivePrefix = {arXiv}, eprint = {1810.04805}, timestamp = {Tue, 30 Oct 2018 20:39:56 +0100},biburl = {https://dblp.org/rec/journals/corr/abs-1810-04805.bib}, bibsource = {dblp computer science bibliography, https://dblp.org}}
2. @inproceedings{wolf-etal-2020-transformers, title = "Transformers: State-of-the-Art Natural Language Processing",author = "Thomas Wolf and Lysandre Debut and Victor Sanh and Julien Chaumond and Clement Delangue and Anthony Moi and Pierric Cistac and Tim Rault and Rémi Louf and Morgan Funtowicz and Joe Davison and Sam Shleifer and Patrick von Platen and Clara Ma and Yacine Jernite and Julien Plu and Canwen Xu and Teven Le Scao and Sylvain Gugger and Mariama Drame and Quentin Lhoest and Alexander M. Rush", booktitle = "Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations", month = oct, year = "2020", address = "Online", publisher = "Association for Computational Linguistics", url = "https://www.aclweb.org/anthology/2020.emnlp-demos.6",pages = "38--45"}