ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
(СПбГУ)

Образовательная программа бакалавриата «Науки о данных»

Отчёт по практике

Учебная практика (научно-исследовательская работа)

Выполнил студент 3 курса бакалавриата
(группа 22.Б05-мкн)
Михеев Артём Антонович

Научный руководитель:
Кудряшов Александр

Санкт-Петербург

Построение обратимых эмбеддингов предложений для финансовых новостей

Содержание

Введение

Обращать эмбеддинги токенов довольно просто. Находим эмбеддинг в матрице эмбеддингов и номер строчки является порядковым номером токена. Используя токенизатор. декодируем и получаем символы.

Предложений бывает очень много, и никакой памяти не хватит для хранения всех эмбедингов. При этом условная обратимость бывает полезной(обратить полностью не получится, так как какая-то часть информации теряется, но получить схожее по смыслу хотелось бы).

Предположим, мы проектируем мультимодальную модель для обработки данных разной природы. К примеру, такая модель может на основе финансовых новостей предсказывать различные показатели: индексы фондового рынка, курс валют. Или генерировать текст - объяснение, почему произошло изменение этих финансовых показателей. Нам важно уловить общий смысл, получить одно представление (вектор) этого объяснения. Из него мы хотим получать текст, не углубляясь на уровень токенов.

Основные результаты практики

Наша задача спроектировать и обучить как энкодер для получения одного представления, так более важная часть модели - decoder для смыслового восстановления.

Dataset.

Будем обучаться на финансовых новостях **Financial News Headlines Data с kaggle.**

Заголовки этих наборов данных, взятые с официальных сайтов CNBC, The Guardian и Reuters, отражают обзор экономики и фондового рынка США за каждый день за 2018-2020.

Пример:

U.S. economy faces significant risks, long road to recovery: IMF staff.

UBS, Morgan Stanley expected to lead Vodafone Tower IPO: sources.

As big U.S. banks let customers delay payments, loan losses remain unclear.

China says it will act to protect its interests after UK Huawei ban.

Всего 53370 заголовков. Делим данные на train и validation, validation size = 0.1.

Метрики.

Надо оценивать на сколько декодируемый текст похож на оригинал. Будем оценивать следующие метрики:

1)Совпадение слов.

ACCURACY. Сопоставляем токены по позициям.

2)Семантическая близость.

BERTScore.

Модель:`'microsoft/deberta-xlarge-mnli'`

- Получает эмбеддинги для каждого токена из двух предложений.
- Вычисляет семантическое соответствие токенов, сопоставляя каждый токен из output с ближайшим по смыслу токеном в input (и наоборот). O(T^2).
- Итоговые P, R, F1 — агрегированные значения сходства между токенами.

bert_score(**"The car was damaged in the accident."**, **"The vehicle got wrecked during the crash."**) = **0.91**

bert_score(**"The economy is growing rapidly."**, **'My dog loves running in the park."**) = **0.35**

*PARAPH-SIM***.**

Модель:`'sentence-transformers/paraphrase-mpnet-base-v2'`

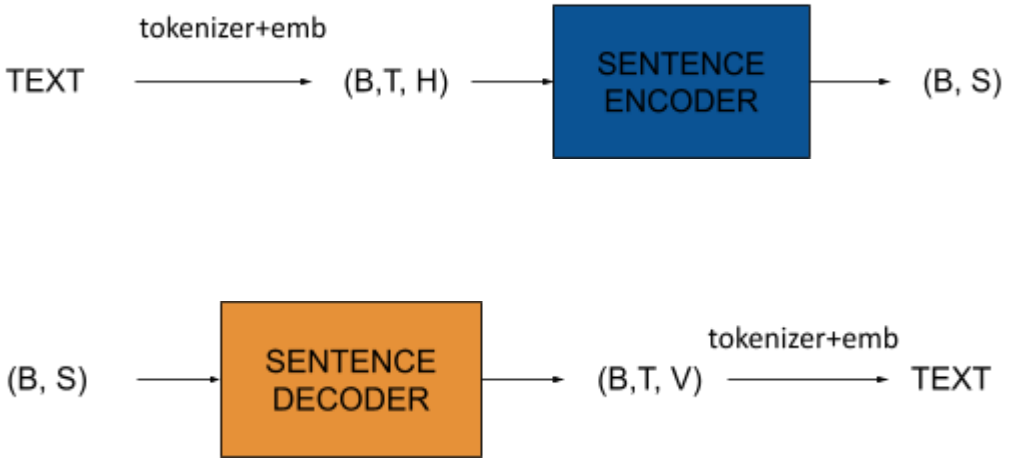
- Каждое предложение сжимается в один вектор фиксированной размерности 768.
- Вычисляется косинусное сходство между двумя векторами:
$$\cos\_\text{sim}(A, B) = \frac{A \cdot B}{\|A\| \cdot \|B\|}$$

paraph_sim(**"The car was damaged in the accident."**, **"The vehicle got wrecked during the crash."**) = **0.821**

paraph_sim(**"The economy is growing rapidly."**, **'My dog loves running in the park."**) = **0.17**

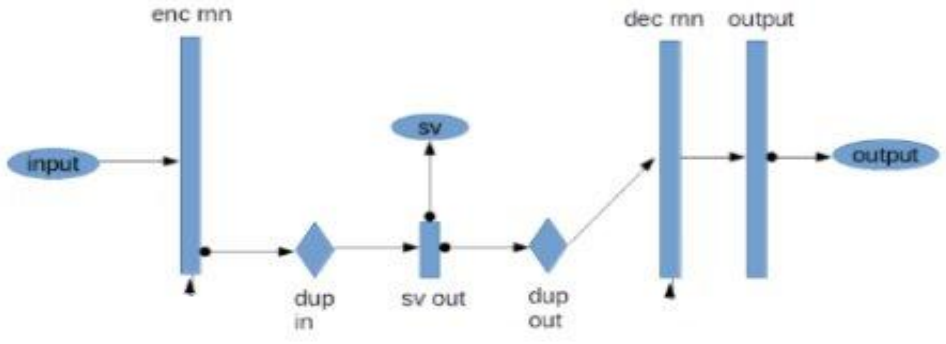
Размерности.

B - batch_size
T - длина предложения в токенах с padding для обучения по бачам.
H - размерность эмбеддингов токенов.
S - размерность эмбеддингов предлодений.
V - размер словаря из токенов.
На вход модели подаётся тензор размера (B, T, H). На выходе logits (B,T,V)



Архитектура моделей.

Прошлые подходы на основе RNN показали способность учиться, однако качество по метрикам было не очень.



Новые подходы на основе transformer архитектуры.

За основу encoder был взят ***FinBERT*** — это предварительно обученная модель для анализа настроений (сентимента) финансовых текстов.

На выходе ***FinBERT*** эмбеддинги для каждого токена и **cls** токен.

Агрегируем с обучаемыми весами выход ***FinBERT*** с помощью ***Attention pooling*** слоя в эмбеддинг предложения. Также для эксперимента подаем только **cls**.

Decoder обучаем с нуля двумя способами.

(1) За один проход. Decoder на основе transformer_encoder.
sent_emb: эмбеддинг предложения. Расширяем до T, учитывая позиции (прибавляем позиционные эмбеддинги) и подаем в decoder для self-attention.
Предсказываем все токены на основе ***sent_emb*.**
P(***tgt***[1:T] | ***sent_emb***)

(2) Авторегрессионно. Decoder на основе transformer_decoder.
***sent_emb*:** эмбеддинг предложения. Подаем в decoder как вход для cross-attention.
tgt: токены входа. Подаем в decoder для masked self-attention и cross-attention.
Предсказываем следующий токен на основе предыдущих и sent_emb.
P(***tgt***[N+1] | ***tgt***[1:N], ***sent_emb***)

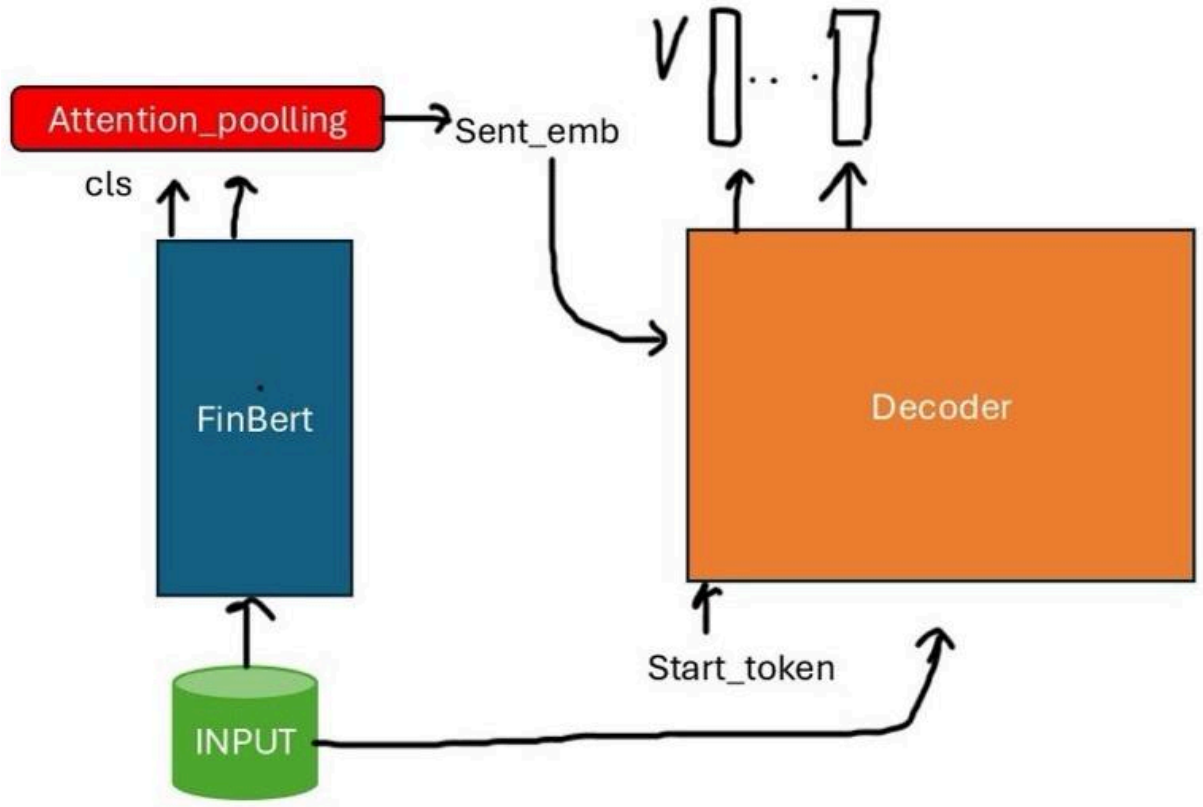
Attention Pooling

```
class AttentionPooling(nn.Module):
    def __init__(self, hidden_dim, output_dim):
        super().__init__()
        self.attn = nn.Linear(hidden_dim, 1)
        self.proj = nn.Sequential(
            nn.Linear(hidden_dim, output_dim),
            nn.GELU(),
            nn.LayerNorm(output_dim),
        )
```

```
    def forward(self, x, mask=None):
        attn_scores = self.attn(x).squeeze(-1)  # (B, T)
        if mask is not None:
            attn_scores = attn_scores.masked_fill(mask == 0, -1e9)
        attn_weights = torch.softmax(attn_scores, dim=1)  # (B, T)
        return torch.sum(attn_weights.unsqueeze(-1) * self.proj(x),
dim=1)  # (B, S)
```

Последний слой для получения ***logits.***

```
self.fc_out = nn.Sequential(
            nn.Dropout(0.5),
            nn.Linear(sent_dim, vocab_size)
        )
```



Обучение

Loss: ***cross_entropy_loss***
+ fine_tuning with ***sent_sim_loss***=cos(sent_emb(input), sent_emb(decoded))

optimizer: AdamW (`learning_rate=0.0001, weight_decay=0.01`)

lr_scheduler: ReduceLROnPlateau (`mode='max', factor=0.1, min_lr=1e-5, patience=5`)

batch_size: 128-200

Возможные улучшения. Loss на уровне предложений

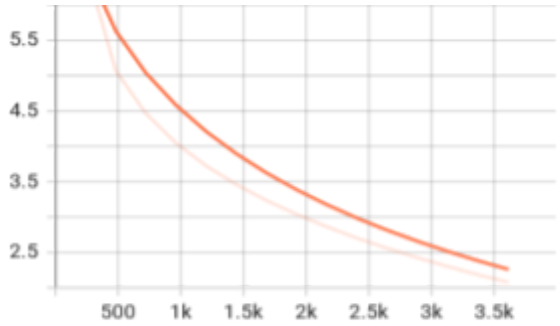
1) sent_sim_loss для эмбедингов из предобученных моделей, как `'paraphrase-mpnet-base-v2'` из paraph_sim, вместо нашей модели.

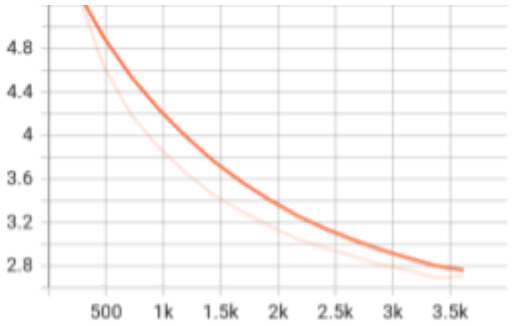
2) Вместо входных токенов в ***cross_entropy_loss*** можно подавать токены перефраз полученных из `'paraphrase-mpnet-base-v2'` в количестве равном T (обрезка или padding до тензора BxT). Это поможет восстанавливать перефразы.

Графики

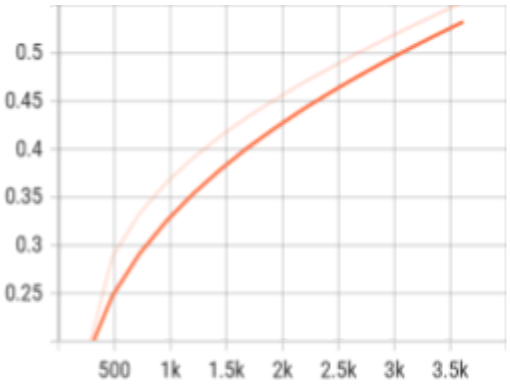
Train *VALIDATION*

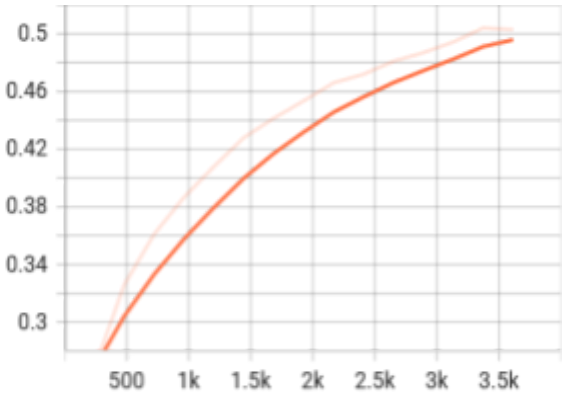
Loss



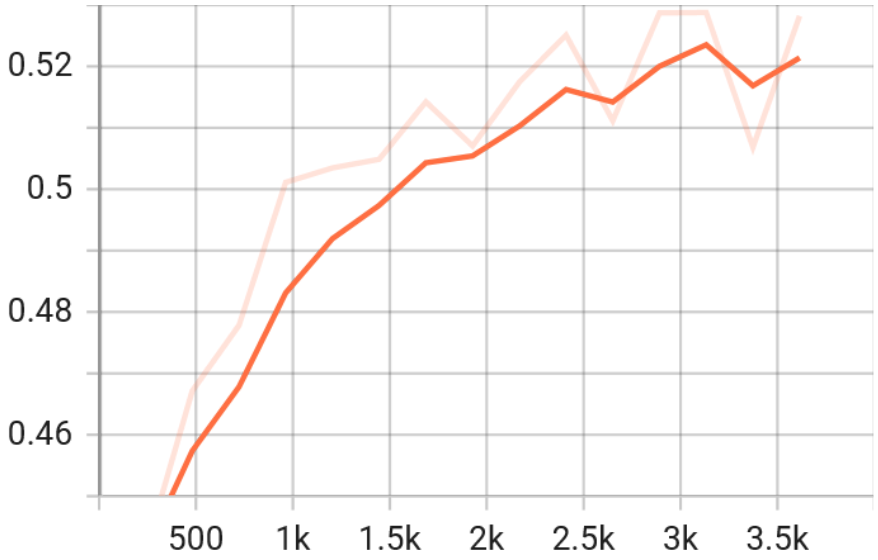


Accuracy

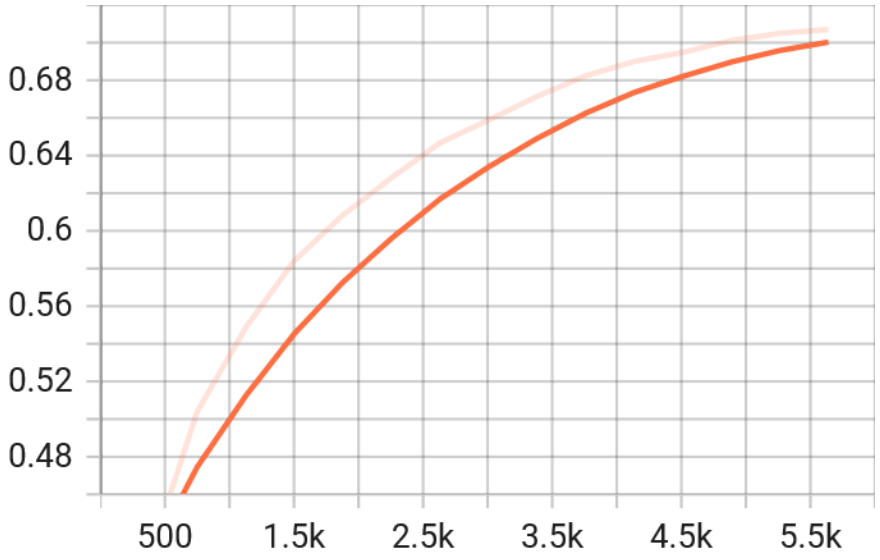




BERT-score



Top-5



Результаты

Зависимость метрик от архитектуры и размера эмбеддинга предложения.

model	*dim*	*Valid_accuracy*	*BERT_score*
За один проход	*1568*	*0.357*	
Авторегрессионно CLS	*768*	*0.406*	*0.58*
Авторегрессионно attention_pooliing	*768*	*0.502*	*0.65*
	2560	*0.552*	*0.675*

Аторегрессионные модели по качеству сильно лучше, однако медленнее.

Attention pooling увеличивает качеству по сравненению с ***cls*** эмбделлингом. Так же этот слой позволяет менять размерность ***sent_emb***.

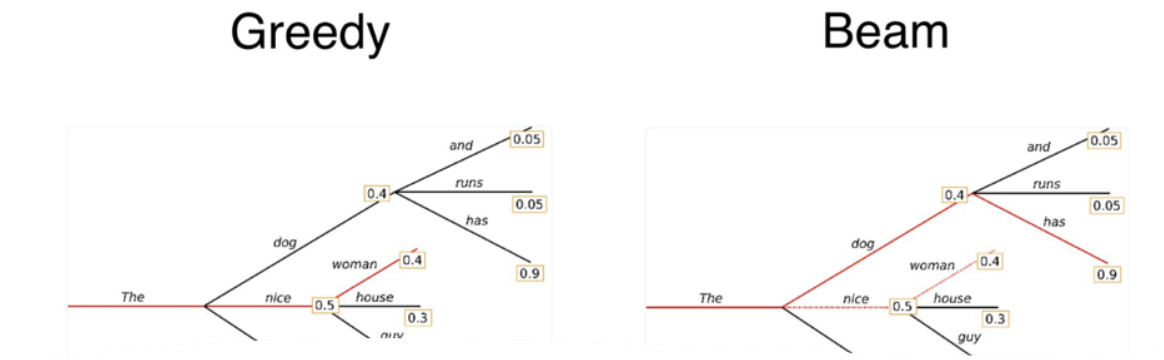
Увеличивая размерность вектора, метрики растут.

Как часто входные токены в первых k токенах по вероятности.

Top-k	*1*	*3*	*5*	*10*	*15*	*25*	*50*	*100*
Model_768	*0.50*	*0.63*	*0.68*	*0.73*	*0.75*	*0.78*	*0.81*	*0.84*
Model-2560	*0.55*	*0.68*	*0.72*	*0.77*	*0.79*	*0.81*	*0.84*	*0.86*

Заметно, что top-5 сильно лучше, чем top-1. С увеличением k скорость роста метрик сильно уменьшается. Возможно, что top-k для маленького k содержит синонимы или перефразы.

Генерация. GREEDY vs BEAM-SEARCH



Изменив алгоритм генерации с жадного на поиск наиболее вероятной последовательности, модель стала реже заикаться (повторять слова).

bert_score: 0.66 → 0.7

paraph_sim: 0.64 → 0.69

Примеры.

INPUT	*DECODED*
consumer giants spurn risks to chase online subscribers	*consumer giants spurn consumer risks to online subscribers*
t - mobile beats estimates for phone subscribers'	*t - mobile beats estimates for phone subscribers*
australia watchdog says vodafone misled customers over digital purchases	*australia watchdog says hsbc over wireless firms misled customers payments*
bayer says monsanto likely kept files on influential people across europe	*bayer says monsanto likely put on some countries sick people in europe*
wall street rises on economic data, easing trade worries	*wall street rises on trade worries, fed data data*
councils to bid for share of fund to invigorate english high streets	*councils to buy out bid to fund for lse's shares of londons*

Заголовки (1-2) целиком восстанавливаются. Остальные примеры имеют похожие слова. В заголовке (3) компания Vodofone восстановлена как фирма, занимающаяся беспроводной связью. Payments почти синоним purchases. В (6) примере english восстанавливается как londons.

Заключение

Основные цели проекта были достигнуты. Была показана возможность сжимать текст в один эмбедддинг с возможностью частичного восстановления с BERTScore=0.7. Эмбеддинги токенов из FinBert или других моделей имеют общую информацию, это позволило нам сжимать новостные заголовки в среднем, состоящие из 30 токенов в один вектор размерности 2560. 30x768 ->2560, сжатие в 10 раз.

Можно добавить *loss* на уровне предложений, чтобы улучшить сжимаемость за счёт перефраз. Увеличить размерность вектора, или чуть изменить архитектуру, например добавить специальные прототокены как в статье[2]. PEFT для FinBert, использовать предобученные *LLM(Llama, Pythia)* как более мощные decoder-архитектуры.

Данный подход открывает новые подходы мультимодальных моделях. Так для объяснения изменения финансовых показателей можно предсказывать один вектор. И наш декодер получит текст, сохранивший общий смысл.

Список использованных литературных источников
и информационных материалов.

[1] Cramming 1568 Tokens into a Single Vector and Back Again: Exploring the Limits of Embedding Space Capacity

[2] Exploring the Latent Capacity of LLMs for One-Step Text Generation