МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «НОВОСИБИРСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

ФАКУЛЬТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

Кафедра Систем информатики

Направление подготовки 09.06.01 – Информатика и вычислительная техника

ОТЧЕТ

Обучающегося Козубенко Андрея Алексеевича **группы № 22215 курса 4**
(Ф.И.О. полностью)

Тема задания: Построение эмбеддингов фрагментов текста

Новосибирск 2025

Оглавление

Введение

Целью лабораторной работы является **разработка модуля для построения эмбеддингов фрагментов текста** в рамках существующего Django-проекта. Эмбеддинг — это числовое представление текста, позволяющее оценивать смысловую близость между предложениями или документами.

Для достижения цели были поставлены следующие задачи:

- Реализовать функцию **разбиения текста на фрагменты** (`get_chunks`);
- Реализовать функцию **генерации эмбеддингов** (`get_embeddings`), использующую предобученную модель `sentence-transformers/paraphrase-multilingual-mpnet-base-v2`;
- Реализовать функцию **сравнения эмбеддингов** по **косинусному сходству** (`cos_compare`);

Реализация

Модуль реализован в составе Django-приложения в файле `db/api/embedding_utils.py`.

1. Основные функции

1.1. `get_chunks(text, chunk_size=200)`
Разбивает текст на последовательные части (чанки) длиной примерно по 200 слов для дальнейшего анализа.

1.2. `get_embeddings(texts)`
Использует модель `paraphrase-multilingual-mpnet-base-v2` для получения векторных представлений каждого текста или чанка.

1.3. `cos_compare(emb1, emb2)`
Вычисляет **косинусное сходство** между двумя эмбеддингами:

$$\cos(\theta) = \frac{v \cdot w}{|| v ||\cdot|| w ||}$$

Результат принимает значения от −1 до 1 (где 1 — максимальное сходство).

---

2. REST-эндпоинты

Для интеграции функций с Django REST Framework созданы API-методы:

Операция	Метод	URL	Описание
Разбить текст на чанки	POST	`/api/embeddings/chunk/`	Возвращает список фрагментов текста
Получить эмбеддинги	POST	`/api/embeddings/build/`	Возвращает эмбеддинги для переданных текстов
Сравнить эмбеддинги	POST	`/api/embeddings/compare/`	Вычисляет косинусное сходство между двумя векторами

Листинги

Листинг 1 – Функции модуля embedding_utils.py

```python
def get_chunks(text: str, chunk_size: int = 200) -> list[str]:
    """
    Разбивает текст на чанки длиной примерно chunk_size слов.
    """
    words = re.findall( pattern: r'\w+', text)
    chunks = []
    for i in range(0, len(words), chunk_size):
        chunk = ' '.join(words[i:i + chunk_size])
        chunks.append(chunk)
    return chunks


2 usages
def get_embeddings(texts: list[str]) -> np.ndarray:
    """
    Возвращает эмбеддинги для списка текстов (или чанков).
    """
    return model.encode(texts, convert_to_numpy=True)


2 usages
def cos_compare(emb1: np.ndarray, emb2: np.ndarray) -> float:
    """
    Вычисляет косинусное сходство между двумя эмбеддингами.
    """
    emb1 = emb1.reshape(1, -1)
    emb2 = emb2.reshape(1, -1)
    return float(cosine_similarity(emb1, emb2)[0][0])
```

Листинг 2 – Пример REST-эндпоинта

```python
2 usages  new *
@api_view(['POST'])
@permission_classes((AllowAny,))
def build_embeddings(request):
    """
    Получает текст(ы), возвращает эмбеддинги.
    """
    data = json.loads(request.body.decode('utf-8'))
    texts = data.get("texts", [])
    embeddings = get_embeddings(texts)
    return Response({"embeddings": embeddings.tolist()})
```

Заключение

В ходе выполнения лабораторной работы была реализована функциональность по **генерации и сравнению эмбеддингов фрагментов текста** в рамках Django-проекта. Были разработаны и протестированы:

1. Функция разбиения текста на фрагменты (get_chunks);
2. Механизм получения эмбеддингов с использованием модели Sentence-Transformers;
3. Метрика косинусного сходства (cos_compare);
4. REST-эндпоинты для интеграции с системой и тестирования в Postman.

Полученный модуль может быть использован для анализа смысловой близости текстов в корпусе — например, при поиске похожих фраз или автоматическом сопоставлении переводов.

Список литературы

1. Django documentation: https://docs.djangoproject.com/
2. Sentence Transformers Documentation — https://www.sbert.net/
3. REST framework: https://www.django-rest-framework.org/