14. Практическое занятие: Семантика текста и векторный анализ. Исследование семантической глубины

Цель занятия

Изучить методы анализа смысла текста через векторные представления, научиться измерять семантическое расстояние между документами и исследовать понятие «семантической глубины» — способности модели улавливать неявные связи и контекстные оттенки.

Теоретический минимум

1. **Семантический вектор документа:** в отличие от Word2Vec (где вектор есть у каждого слова), здесь мы получаем единый вектор для всего предложения или абзаца (например, через усреднение векторов слов или модель **Doc2Vec**).
2. **Косинусное сходство (Cosine Similarity):** основной инструмент измерения близости смыслов. Значение 1.0 — смыслы идентичны, 0 — связи нет.
3. **Семантическая глубина:** характеристика модели, определяющая, насколько она учитывает переносные значения, идиомы и контекст. Глубокие модели (трансформеры) различают «ключ» (от замка) и «ключ» (родник).

Задание 1. Центроидный метод (Усреднение векторов)

Самый простой способ получить семантический вектор текста — вычислить среднее арифметическое всех векторов входящих в него слов.

Инструкция:

Реализуйте функцию получения вектора предложения на основе предобученной модели Word2Vec.

```
import numpy as np

def get_sentence_vector(sentence, model):
    words = [w for w in sentence.lower().split() if w in model]
    if not words:
        return np.zeros(model.vector_size)
    # Усредняем векторы всех найденных слов
    word_vectors = [model[w] for w in words]
    return np.mean(word_vectors, axis=0)

# Пример
vec1 = get_sentence_vector("искусственный интеллект развивает науку",
model)
print(f"Размерность вектора предложения: {vec1.shape}")
```

Задание 2. Анализ семантического расстояния

Математический анализ позволяет найти скрытую близость между текстами, которые не имеют общих слов, но близки по смыслу.

Инструкция:

Сравните три предложения и докажите, что предложения 1 и 2 семантически ближе друг к другу, чем к предложению 3.

```
from sklearn.metrics.pairwise import cosine_similarity

s1 = "космический корабль летит к звездам"
s2 = "астронавты изучают далекие галактики"
s3 = "вчера я приготовил вкусный ужин"

v1 = get_sentence_vector(s1, model).reshape(1, -1)
v2 = get_sentence_vector(s2, model).reshape(1, -1)
v3 = get_sentence_vector(s3, model).reshape(1, -1)

print(f"Сходство (Космос vs Астронавты): {cosine_similarity(v1,
v2)[0][0]:.3f}")
print(f"Сходство (Космос vs Ужин): {cosine_similarity(v1, v3)[0][0]:.3f}")
```

Задание 3. Исследование семантической глубины (Идиомы)

Проверьте, понимает ли векторная модель «глубину» фразеологизмов.

Инструкция:

Сравните векторы для слов «лодырь» и «бить баклуши».

1. Если векторы близки — модель обладает семантической глубиной.
2. Если модель выдает низкое сходство — она опирается на буквальное значение слов (бить, баклуши).

```
# Задание для самостоятельной проверки:
# Попробуйте пары: "медленно" / "черепашьим шагом", "обманывать" / "водить
за нос"
```

Задание 4. Построение семантического ландшафта (Heatmap)

Визуализация матрицы сходства позволяет увидеть, как разные части текста коррелируют между собой.

Инструкция:

Постройте тепловую карту сходства для набора из 5 предложений разной тематики.

```
import seaborn as sns
import matplotlib.pyplot as plt

sentences = [
    "Физика изучает законы природы",
    "Биология исследует живые организмы",
    "Математика — язык науки",
    "Повар готовит завтрак",
    "Рецепт пирога очень сложный"
]

vectors = [get_sentence_vector(s, model) for s in sentences]
sim_matrix = cosine_similarity(vectors)

sns.heatmap(sim_matrix, annot=True, xticklabels=False,
yticklabels=sentences, cmap="YlGnBu")
plt.title("Семантическое сходство документов")
plt.show()
```

Задание 5. Анализ контекстной неоднозначности

Объясните лингвистический парадокс: почему в старых моделях (Word2Vec) слова «хороший» и «плохой» часто имеют очень высокую семантическую близость?

Подсказка: подумайте о контекстах, в которых они встречаются («фильм был очень ...»).

Контрольные вопросы

1. Как «семантическая глубина» помогает в работе поисковых систем (поиск не по словам, а по смыслу)?
2. Почему простое усреднение векторов плохо работает для очень длинных текстов (целых книг)?
3. В чем преимущество взвешенного усреднения (например, с весами TF-IDF) над обычным?

Итог работы

Вы научились проводить глубокий векторный анализ текстов, выходя за рамки простого поиска совпадений слов. Эти методы позволяют оценивать качество переводов, группировать новости по сюжетам и создавать рекомендательные системы на основе интересов пользователей.