13. Практическое занятие: Векторизация слов и семантические пространства (Word2Vec, GloVe)

Цель занятия

Изучить концепцию дистрибутивной семантики, научиться переходить от разреженных векторов (TF-IDF) к плотным векторным представлениям (Embeddings) и анализировать смысловые связи между словами через математические операции.

Теоретический минимум

Word Embeddings (Вложения слов) — это способ представления слов в виде векторов фиксированной размерности (обычно от 100 до 300), где близкие по смыслу слова имеют близкие координаты.

1. **Дистрибутивная гипотеза:** «Слово познается по его окружению». Слова, встречающиеся в схожих контекстах, имеют схожие значения.
2. **Word2Vec:** Модель (от Google), использующая нейросетевой подход для обучения векторов (архитектуры CBOW и Skip-gram).
3. **GloVe (Global Vectors):** Модель (от Stanford), основанная на глобальной статистике совместной встречаемости слов в корпусе.

Задание 1. Загрузка предобученной модели Word2Vec

Обучение качественной модели требует огромных корпусов (миллиарды слов). На практике часто используют готовые модели, например, из проекта RusVectores.

Инструкция:

Используйте библиотеку gensim для загрузки модели и поиска синонимов.

```
import gensim.downloader as api

# Загрузка небольшой модели (например, обученной на Wikipedia)
# Примечание: загрузка может занять время
model = api.load("word2vec-ruscorpora-300")

# Поиск слов, близких по смыслу к слову "компьютер"
# Важно: в моделях RusVectores нужно указывать часть речи: слово_POS
word = "компьютер_NOUN"
if word in model:
    similar_words = model.most_similar(word, topn=5)
    for w, score in similar_words:
        print(f"{w:<20} | Сходство: {score:.2f}")
```

Задание 2. Семантическая арифметика

Одна из самых известных особенностей Word2Vec — способность решать пропорции вида «A относится к B так же, как C относится к D».

Инструкция:

Выполните классическое вычисление: Король - Мужчина + Женщина = ?

```
# Математическая операция над векторами: king - man + woman
result = model.most_similar(positive=['король_NOUN', 'женщина_NOUN'],
negative=['мужчина_NOUN'])

print("Результат операции 'Король - Мужчина + Женщина':")
for w, score in result[:3]:
    print(f"{w}: {score:.2f}")
```

Задание 3. Визуализация семантического пространства (t-SNE)

Векторы имеют размерность 300, что невозможно увидеть. Для визуализации используют алгоритмы снижения размерности, такие как t-SNE или PCA, чтобы спроецировать их на плоскость (2D).

Инструкция:

Выберите 3 группы слов (например: «фрукты», «города», «инструменты») и отобразите их на графике.

```
from sklearn.manifold import TSNE
import matplotlib.pyplot as plt
import numpy as np

words = ["яблоко_NOUN", "груша_NOUN", "слива_NOUN",
         "москва_NOUN", "париж_NOUN", "берлин_NOUN",
         "молоток_NOUN", "пила_NOUN", "топор_NOUN"]

# Извлекаем векторы
word_vectors = np.array([model[w] for w in words])

# Снижаем размерность до 2D
tsne = TSNE(n_components=2, perplexity=5, random_state=42)
vectors_2d = tsne.fit_transform(word_vectors)

# Визуализация
plt.figure(figsize=(10, 8))
for i, word in enumerate(words):
    plt.scatter(vectors_2d[i, 0], vectors_2d[i, 1])
    plt.text(vectors_2d[i, 0] + 0.1, vectors_2d[i, 1] + 0.1, word)
plt.title("Визуализация семантических кластеров")
plt.show()
```

Задание 4. Анализ лишнего слова (Odd-One-Out)

Модели эмбеддингов позволяют находить слово, которое не вписывается в логический ряд.

Инструкция:

Используйте метод doesnt_match, чтобы найти лишнее слово в списке: завтрак, обед, ужин, компьютер.

```
list_to_check = ["завтрак_NOUN", "обед_NOUN", "ужин_NOUN",
"компьютер_NOUN"]
odd_word = model.doesnt_match(list_to_check)
print(f"Лишнее слово в списке: {odd_word}")
```

Задание 5. Сравнение Word2Vec и GloVe (Теоретическое)

Изучите различия в подходах. Word2Vec «смотрит» на локальный контекст (окно слов), а GloVe учитывает глобальную матрицу совместной встречаемости слов во всем корпусе.

Контрольное задание:

Напишите, какой метод будет лучше работать на маленьком специализированном корпусе текстов (например, медицинских картах), а какой — на гигантском дампе интернета? Обоснуйте ответ.

Контрольные вопросы

1. Почему эмбеддинги (плотные векторы) лучше подходят для нейронных сетей, чем One-Hot Encoding?
2. Что произойдет, если мы попытаемся найти вектор слова, которого не было в обучающем словаре (проблема OOV — Out of Vocabulary)?
3. Как размер «окна» (количество слов слева и справа) влияет на то, какие связи выучит модель?

Итог работы

Студенты перешли от формального анализа символов к работе со смысловыми векторами. Понимание эмбеддингов является «мостиком» к современным языковым моделям (BERT, GPT), где векторы становятся контекстно-зависимыми.