Лабораторна робота №2. Методи векторизації тексту

Мета: засвоїти основні методи векторизації тексту та word embedding

Теоретичні відомості

Count Vectorizer. Перетворіть набір текстових документів у матрицю кількості маркерів.

Приклад 1. Застосування з бібліотеки sklearn.

```
# create the transform

vectorizer = CountVectorizer()

# tokenizing

vectorizer.fit(text)

# encode document

vector = vectorizer.transform(text)
```

	I	love	NLP	is	future	will	learn	In	2month
I love NLP and I will learn NLP in 2 months	2	1	2	0	0	1	1	1	1
NLP is future	0	0	1	1	1	0	0	0	0

```
Result:

{'love': 4, 'nlp': 5, 'and': 1, 'will': 6, 'learn': 3, 'in': 2, '2month': 0}
[[1 1 1 1 1 2 1]]
```

Count Vectorizer та n-грами. Це поєднання декількох символів або слів при використанні Count Vectorizer.

Приклад 2. Count Vectorizer та n-грами

```
from sklearn.feature_extraction.text import CountVectorizer

# Text

text = ["I love NLP and I will learn NLP in 2month "]

# create the transform

vectorizer = CountVectorizer(ngram_range=(2,2))

# tokenizing

vectorizer.fit(text)

# encode document

vector = vectorizer.transform(text)
```

Hash Vectorizing. Хеш функція використовується для кодування слів у попередніх варіантах векторизації.

Term frequency-Inverse document frequency (TF-IDF).

TF фіксує важливість слова незалежно від довжини документа

TF(t) = (Кількість разів термін t з'являється в документі) /

(Загальна кількість термінів у документі)

IDF вимірює рідкість терміна. Слова на кшталт артиклів (англ.) з'являються у всьому корпусі документів. Отже, якщо слово з'являється в майже У всіх документах це слово не має користі, оскільки воно не допомагає з класифікацією.

$$IDF = \log(N/n),$$

де N – загальна кількість документів, а n – кількість документів, у яких слово було присутнім.

$$TF\text{-}IDF = TF * IDF$$

Word embeddings. Це техніка навчання ознак тексту, при якій словниковий запас відображається у вектори дійсних чисел, відображаючи контекстну ієрархію та семантичну близькість.

- Word2vec (Google)
- Bert (Google)
- Glove (Stanford)
- FastText (Facebook's AI Research Lab)
- ELMo (Allen Institute for Artificial Intelligence)

Використовують попередньо навчені нейронні мережі та переніс навчання.

Приклад 3. Бібліотека gensim

```
!pip install gensim

import gensim
from gensim.models import Word2Vec
```

Завдання для самостійної роботи

1. Використовуючи приклад https://www.kaggle.com/avk256/lab2-ex виконати векторизацію набору даних, отриманих в попередній роботі, використовуючи методи Count Vectorizer, TFIDF.

2. Застосувати метод Word2vec з прикладу до набору даних Spooky

3. Порівняти отримані результати та проаналізувати, який з методів векторизації більш перспективний для подальшої класифікації текстів.

Запитання для самоконтролю

1. В чому різниця між методами word vecrorising та word embedding?

2. Переваги та недоліки методу **TF-IDF**.

3. Що таке переніс навчання та як він використовується в методах word embedding?