Введение в обработку текста на естественном языке

Материалы:

- Макрушин С.В. Лекция 9: Введение в обработку текста на естественном языке\
- https://realpython.com/nltk-nlp-python/
- https://scikit-learn.org/stable/modules/feature_extraction.html

Задачи для совместного разбора

In [46]:
```
from sklearn.feature_extraction.text import CountVectorizer
import pymorphy2
```

1. Считайте слова из файла `litw-win.txt` и запишите их в список `words`. В заданном предложении исправьте все опечатки, заменив слова с опечатками на ближайшие (в смысле расстояния Левенштейна) к ним слова из списка `words`. Считайте, что в слове есть опечатка, если данное слово не содержится в списке `words`.

In [49]:
```
text = '''с велечайшим усилием выбравшись из потока убегающих людей Кутузов со свитой уменьшевшейся вдвое поеха
```

2. Разбейте текст из формулировки задания 1 на слова; проведите стемминг и лемматизацию слов.

3. Преобразуйте предложения из формулировки задания 1 в векторы при помощи `CountVectorizer`.

Лабораторная работа 9

In [54]:
```
import pandas as pd
import random
import re
import numpy as np
import nltk
from nltk.metrics import edit_distance
from nltk.tokenize import word_tokenize
from nltk.metrics import edit_distance
from nltk.corpus import stopwords
from nltk.stem import SnowballStemmer, WordNetLemmatizer
from collections import Counter
from sklearn.feature_extraction.text import TfidfVectorizer
from scipy.spatial.distance import cosine

nltk.download('stopwords')
nltk.download('punkt')
nltk.download('wordnet')
```

```
[nltk_data] Downloading package stopwords to
[nltk_data]     /Users/egorsipilov/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
[nltk_data] Downloading package punkt to
[nltk_data]     /Users/egorsipilov/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package wordnet to
[nltk_data]     /Users/egorsipilov/nltk_data...
[nltk_data]   Package wordnet is already up-to-date!
```

Out[54]: True

Расстояние редактирования

1.1 Загрузите предобработанные описания рецептов из файла `recipes_sample.csv`. Получите набор уникальных слов `words`, содержащихся в текстах описаний рецептов (воспользуйтесь `word_tokenize` из `nltk`).

In [58]:
```
recipes = pd.read_csv('recipes_sample.csv')

descriptions = recipes['description'].dropna().str.lower()


words = set()
for desc in descriptions:
    tokens = word_tokenize(desc)
    words.update(w for w in tokens if re.match(r'^[a-zA-Z]+$', w))
```

```
list(words)[:20]
```

Out[58]:
```
['needles',
 'hussard',
 'munchers',
 'grasp',
 'moroccan',
 'cuates',
 'thrilling',
 'preventing',
 'woodruff',
 'mocktails',
 'sofrito',
 'shelters',
 'nimz',
 'cork',
 'prematurely',
 'arugula',
 'heh',
 'eggrolls',
 'addicts',
 'sandwich']
```

In [60]:
```
len(words)
```

Out[60]: 21949

1.2 Сгенерируйте 5 пар случайно выбранных слов и посчитайте между ними расстояние редактирования.

In [63]:
```
random.seed(42)
word_list = list(words)
pairs = [random.sample(word_list, 2) for _ in range(5)]

for i, (word1, word2) in enumerate(pairs, 1):
    distance = edit_distance(word1, word2)
    print(f"Пара {i}: '{word1}' и '{word2}' -> Расстояние редактирования: {distance}")
```

```
Пара 1: 'absorbs' и 'absorbing' -> Расстояние редактирования: 3
Пара 2: 'crinkly' и 'damian' -> Расстояние редактирования: 7
Пара 3: 'zaarite' и 'gentile' -> Расстояние редактирования: 5
Пара 4: 'curies' и 'andrea' -> Расстояние редактирования: 5
Пара 5: 'behavior' и 'tackling' -> Расстояние редактирования: 7
```

1.3 Напишите функцию, которая для заданного слова `word` возвращает `k` ближайших к нему слов из списка `words` (близость слов измеряется с помощью расстояния Левенштейна)

In [66]:
```
def find_closest_words(word, words_list, k=5):
    distances = [(w, edit_distance(word, w)) for w in words_list]
    distances.sort(key=lambda x: x[1])
    return distances[:k]
```

In [68]:
```
test_word = "food"
closest = find_closest_words(test_word, words, k=5)


for w, dist in closest:
    print(f"Слово: {w}, Расстояние: {dist}")
```

```
Слово: food, Расстояние: 0
Слово: fold, Расстояние: 1
Слово: fond, Расстояние: 1
Слово: ford, Расстояние: 1
Слово: flood, Расстояние: 1
```

Стемминг, лемматизация

2.1 На основе результатов 1.1 создайте `pd.DataFrame` со столбцами: * word * stemmed_word * normalized_word

Столбец `word` укажите в качестве индекса.

Для стемминга воспользуйтесь `SnowballStemmer`, для нормализации слов - `WordNetLemmatizer`. Сравните результаты стемминга и лемматизации.

In [72]:
```
stemmer = SnowballStemmer("english")
lemmatizer = WordNetLemmatizer()


data = {
    'word': list(words),
    'stemmed_word': [stemmer.stem(w) for w in words],
    'normalized_word': [lemmatizer.lemmatize(w) for w in words]
```

```
}


df_words = pd.DataFrame(data).set_index('word')

df_words.head(15)
```

Out[72]:

word	stemmed_word	normalized_word
needles	needl	needle
hussard	hussard	hussard
munchers	muncher	muncher
grasp	grasp	grasp
moroccan	moroccan	moroccan
cuates	cuat	cuates
thrilling	thrill	thrilling
preventing	prevent	preventing
woodruff	woodruff	woodruff
mocktails	mocktail	mocktails
sofrito	sofrito	sofrito
shelters	shelter	shelter
nimz	nimz	nimz
cork	cork	cork
prematurely	prematur	prematurely

2.2. Удалите стоп-слова из описаний рецептов. Какую долю об общего количества слов составляли стоп-слова? Сравните топ-10 самых часто употребляемых слов до и после удаления стоп-слов.

In [75]:
```
stop_words = set(stopwords.words('english'))


all_words = []
for desc in recipes['description'].dropna().str.lower():
    wd = word_tokenize(desc)
    all_words.extend(wd)


total_words = len(all_words)
stop_words_count = sum(1 for w in all_words if w in stop_words)
stop_words_proportion = stop_words_count / total_words

word_cnt_before = Counter(all_words).most_common(10)

filtered_tokens = [w for w in all_words if w not in stop_words]

word_cnt_after = Counter(filtered_tokens).most_common(10)

print(f"Доля стоп слов: {stop_words_proportion:.2%}")
```

```
Доля стоп слов: 40.27%
```

In [77]:
```
for word, count in word_cnt_before:
    print(f"'{word}': {count}")
```

```
'.': 66166
'the': 40257
',': 38544
'a': 35030
'and': 30425
'i': 27799
'this': 27132
'to': 23508
'it': 23212
'is': 20501
```

In [79]:
```
for word, count in word_cnt_after:
    print(f"'{word}': {count}")
```

```
'.': 66166
',': 38544
'!': 16054
'recipe': 15122
''s': 7688
'make': 6367
'time': 5198
'n't': 4798
'use': 4645
')': 4587
```

Векторное представление текста

3.1 Выберите случайным образом 5 рецептов из набора данных. Представьте описание каждого рецепта в виде числового вектора при помощи `TfidfVectorizer`

In [83]:
```
random.seed(42)
sample_recipes = recipes.dropna(subset=['description']).sample(5, random_state=42)

vectorizer = TfidfVectorizer()

tfidf_matrix = vectorizer.fit_transform(sample_recipes['description'])


tfidf_df = pd.DataFrame(
    tfidf_matrix.toarray(),
    index=sample_recipes['name'],
    columns=vectorizer.get_feature_names_out()
)
```

In [85]:
```
tfidf_matrix.toarray()
```

Out[85]:
```
array([[0.        , 0.        , 0.        , 0.        , 0.        ,
        0.13129327, 0.        , 0.15816745, 0.        , 0.        ,
        0.        , 0.        , 0.19604448, 0.19604448, 0.        ,
        0.        , 0.        , 0.        , 0.19604448, 0.        ,
        0.        , 0.        , 0.        , 0.19604448, 0.        ,
        0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        , 0.19604448, 0.19604448,
        0.        , 0.19604448, 0.        , 0.19604448, 0.        ,
        0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.31633491, 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        , 0.        , 0.15816745,
        0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        , 0.        , 0.19604448,
        0.        , 0.        , 0.        , 0.19604448, 0.        ,
        0.19604448, 0.        , 0.        , 0.        , 0.        ,
        0.19604448, 0.09341625, 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        , 0.        , 0.        ,
        0.11044804, 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.15816745, 0.19604448, 0.        , 0.19604448,
        0.19604448, 0.        , 0.        , 0.        , 0.        ,
        0.26258655, 0.        , 0.09341625, 0.13129327, 0.        ,
        0.19604448, 0.19604448, 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        , 0.        , 0.        ,
        0.19604448, 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        ],
       [0.11327136, 0.11327136, 0.        , 0.        , 0.09138662,
        0.3034366 , 0.11327136, 0.        , 0.11327136, 0.11327136,
        0.22654272, 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.11327136, 0.11327136, 0.        , 0.11327136,
        0.        , 0.11327136, 0.        , 0.        , 0.11327136,
        0.        , 0.        , 0.11327136, 0.11327136, 0.        ,
        0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.09138662, 0.11327136, 0.        ,
        0.        , 0.        , 0.11327136, 0.        , 0.        ,
        0.11327136, 0.        , 0.        , 0.        , 0.09138662,
        0.        , 0.        , 0.        , 0.18277324, 0.06381511,
        0.11327136, 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.11327136, 0.18277324, 0.        ,
        0.        , 0.11327136, 0.        , 0.        , 0.11327136,
        0.        , 0.        , 0.        , 0.        , 0.22654272,
        0.        , 0.05397441, 0.11327136, 0.11327136, 0.        ,
        0.11327136, 0.11327136, 0.        , 0.        , 0.11327136,
        0.06381511, 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.09138662, 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        , 0.        , 0.        ,
        0.37929575, 0.        , 0.05397441, 0.        , 0.09138662,
        0.        , 0.        , 0.        , 0.33981408, 0.        ,
        0.22654272, 0.        , 0.        , 0.11327136, 0.11327136,
```

```
 0.        , 0.11327136, 0.        , 0.11327136, 0.        ,
 0.        , 0.        , 0.11327136],
[0.        , 0.        , 0.        , 0.        , 0.        ,
 0.        , 0.        , 0.        , 0.        , 0.        ,
 0.        , 0.        , 0.        , 0.        , 0.        ,
 0.        , 0.        , 0.        , 0.        , 0.        ,
 0.        , 0.        , 0.19843799, 0.        , 0.        ,
 0.        , 0.        , 0.        , 0.        , 0.        ,
 0.19843799, 0.        , 0.16009852, 0.        , 0.        ,
 0.        , 0.        , 0.        , 0.        , 0.        ,
 0.        , 0.        , 0.        , 0.        , 0.        ,
 0.        , 0.16009852, 0.        , 0.        , 0.        ,
 0.        , 0.19843799, 0.        , 0.        , 0.        ,
 0.19843799, 0.16009852, 0.19843799, 0.        , 0.22359301,
 0.        , 0.19843799, 0.        , 0.        , 0.19843799,
 0.        , 0.16009852, 0.        , 0.16009852, 0.        ,
 0.39687599, 0.        , 0.16009852, 0.        , 0.        ,
 0.        , 0.        , 0.        , 0.        , 0.        ,
 0.        , 0.09455677, 0.        , 0.        , 0.        ,
 0.        , 0.        , 0.19843799, 0.        , 0.        ,
 0.1117965 , 0.        , 0.19843799, 0.19843799, 0.19843799,
 0.        , 0.        , 0.        , 0.        , 0.        ,
 0.        , 0.19843799, 0.19843799, 0.19843799, 0.        ,
 0.        , 0.        , 0.09455677, 0.13289624, 0.        ,
 0.        , 0.        , 0.        , 0.        , 0.19843799,
 0.        , 0.        , 0.        , 0.        , 0.        ,
 0.        , 0.        , 0.        , 0.        , 0.        ,
 0.        , 0.        , 0.        ],
[0.        , 0.        , 0.        , 0.        , 0.        ,
 0.41808942, 0.        , 0.        , 0.        , 0.        ,
 0.        , 0.        , 0.        , 0.        , 0.        ,
 0.        , 0.        , 0.        , 0.        , 0.        ,
 0.20809424, 0.        , 0.        , 0.        , 0.        ,
 0.        , 0.20809424, 0.        , 0.        , 0.20809424,
 0.        , 0.20809424, 0.16788912, 0.        , 0.        ,
 0.20809424, 0.        , 0.        , 0.        , 0.        ,
 0.        , 0.        , 0.        , 0.        , 0.        ,
 0.20809424, 0.        , 0.        , 0.        , 0.        ,
 0.        , 0.        , 0.        , 0.20809424, 0.        ,
 0.        , 0.        , 0.        , 0.16788912, 0.11723667,
 0.        , 0.        , 0.20809424, 0.20809424, 0.        ,
 0.16788912, 0.16788912, 0.        , 0.        , 0.        ,
 0.        , 0.        , 0.        , 0.        , 0.        ,
 0.        , 0.        , 0.        , 0.        , 0.        ,
 0.        , 0.09915802, 0.        , 0.        , 0.20809424,
 0.        , 0.        , 0.        , 0.        , 0.        ,
 0.        , 0.        , 0.        , 0.        , 0.        ,
 0.20809424, 0.        , 0.        , 0.20809424, 0.        ,
 0.        , 0.        , 0.        , 0.        , 0.20809424,
 0.        , 0.        , 0.19831604, 0.        , 0.        ,
 0.        , 0.        , 0.        , 0.        , 0.        ,
 0.        , 0.        , 0.        , 0.        , 0.        ,
 0.        , 0.        , 0.20809424, 0.        , 0.20809424,
 0.        , 0.        , 0.        ],
[0.        , 0.        , 0.17259585, 0.17259585, 0.13924925,
 0.        , 0.        , 0.13924925, 0.        , 0.        ,
 0.        , 0.17259585, 0.        , 0.        , 0.17259585,
 0.17259585, 0.        , 0.        , 0.        , 0.        ,
 0.        , 0.        , 0.        , 0.        , 0.        ,
 0.17259585, 0.        , 0.        , 0.        , 0.        ,
 0.        , 0.        , 0.        , 0.        , 0.        ,
 0.        , 0.        , 0.17259585, 0.        , 0.17259585,
 0.17259585, 0.17259585, 0.13924925, 0.        , 0.17259585,
 0.        , 0.        , 0.        , 0.17259585, 0.17259585,
 0.        , 0.        , 0.17259585, 0.        , 0.        ,
 0.        , 0.13924925, 0.        , 0.        , 0.09723749,
 0.        , 0.        , 0.        , 0.        , 0.        ,
 0.13924925, 0.        , 0.        , 0.        , 0.        ,
 0.        , 0.        , 0.13924925, 0.        , 0.        ,
 0.        , 0.17259585, 0.17259585, 0.17259585, 0.        ,
 0.        , 0.08224285, 0.        , 0.        , 0.        ,
 0.        , 0.        , 0.        , 0.17259585, 0.        ,
 0.09723749, 0.17259585, 0.        , 0.        , 0.        ,
 0.        , 0.        , 0.        , 0.        , 0.        ,
 0.        , 0.        , 0.        , 0.        , 0.        ,
 0.23117891, 0.17259585, 0.1644857 , 0.11558946, 0.13924925,
 0.        , 0.        , 0.17259585, 0.        , 0.        ,
 0.        , 0.17259585, 0.17259585, 0.        , 0.        ,
 0.        , 0.        , 0.        , 0.        , 0.        ,
 0.17259585, 0.17259585, 0.        ]])
```

In [87]:
```
tfidf_df.head()
```

Out[87]:

name	add	adjust	ago	all	always	and	are	as	banana	bananas	...	very	wa
never fail blender hollandaise sauce	0.000000	0.000000	0.000000	0.000000	0.000000	0.131293	0.000000	0.158167	0.000000	0.000000	...	0.000000	0.0(
banana bread from betty crocker	0.113271	0.113271	0.000000	0.000000	0.091387	0.303437	0.113271	0.000000	0.113271	0.113271	...	0.113271	0.11
sweet potatoes supreme	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.000000	0.0(
lemon dill chicken patties with orzo	0.000000	0.000000	0.000000	0.000000	0.000000	0.418089	0.000000	0.000000	0.000000	0.000000	...	0.000000	0.0(
rice broccoli casserole w nutritional yeast	0.000000	0.000000	0.172596	0.172596	0.139249	0.000000	0.000000	0.139249	0.000000	0.000000	...	0.000000	0.0(

5 rows × 128 columns

3.2 Вычислите близость между каждой парой рецептов, выбранных в задании 3.1, используя косинусное расстояние (`scipy.spatial.distance.cosine`) Результаты оформите в виде таблицы `pd.DataFrame` . В качестве названий строк и столбцов используйте названия рецептов.

In [90]:
```
recipe_names = sample_recipes['name'].values
n = len(recipe_names)
distance_matrix = np.zeros((n, n))


for i in range(n):
    for j in range(n):
        if i != j:
            distance_matrix[i, j] = cosine(tfidf_matrix[i].toarray()[0], tfidf_matrix[j].toarray()[0])
        else:
            distance_matrix[i, j] = 0.0


distance_df = pd.DataFrame(distance_matrix, index=recipe_names, columns=recipe_names)



distance_df
```

Out[90]:

	never fail blender hollandaise sauce	banana bread from betty crocker	sweet potatoes supreme	lemon dill chicken patties with orzo	rice broccoli casserole w nutritional yeast
never fail blender hollandaise sauce	0.000000	0.814522	0.901893	0.917319	0.868307
banana bread from betty crocker	0.814522	0.000000	0.939128	0.818913	0.848411
sweet potatoes supreme	0.901893	0.939128	0.000000	0.891901	0.884109
lemon dill chicken patties with orzo	0.917319	0.818913	0.891901	0.000000	0.924447
rice broccoli casserole w nutritional yeast	0.868307	0.848411	0.884109	0.924447	0.000000

3.3 Какие рецепты являются наиболее похожими? Прокомментируйте результат (словами).

In [93]:
```
np.fill_diagonal(distance_df.values, np.inf)
min_distance = distance_df.values.min()
min_idx = np.unravel_index(distance_df.values.argmin(), distance_df.shape)
recipe1, recipe2 = distance_df.index[min_idx[0]], distance_df.columns[min_idx[1]]


desc1 = recipes[recipes['name'] == recipe1]['description'].values[0]
desc2 = recipes[recipes['name'] == recipe2]['description'].values[0]
```

```
print(f"Наиболее похожие рецепты: '{recipe1}' and '{recipe2}'")
print(f"Косинусовое расстояние: {min_distance:.4f}")
print(f"\nОписание '{recipe1}':\n{desc1}")
print(f"\nОписание '{recipe2}':\n{desc2}")
```

```
Наиболее похожие рецепты: 'never fail blender hollandaise sauce' and 'banana bread from betty crocker'
Косинусовое расстояние: 0.8145

Описание 'never fail blender hollandaise sauce':
from the best of bridge. this fantastic recipe has changed sunday morning eggs benny from a special occasion tre
at to a weekly tradition! so easy and as the name states-never fail!!!

Описание 'banana bread from betty crocker':
this is the recipe i always use for banana bread.  it is very forgiving and you can add more nuts, more bananas
(just adjust the milk quantity so the batter has the consistency of muffin batter), and only need one bowl.  i u
se walnuts usually, but pecans are good too and i usually use the cheap 'whole and pieces' nuts without chopping
further.
```

Комментарий:

Наиболее похожими рецептами являются 'never fail blender hollandaise sauce' и 'banana bread from betty crocker' с косинусным расстоянием 0.8145, что указывает на высокую схожесть их описаний. Их близость обусловлена акцентом на простоту и удобство приготовления. Это показано в Tfidf векторах, что делает их ближе друг к другу, чем к другим рецептам