Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н. Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н. Э. Баумана)

ФАКУЛЬТЕТ «Информатика, искусственный интеллект и системы управления»

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

ОТЧЕТ

по Лабораторной работе

на тему: «Обработка естественного языка»

Студент ИУ7-13М (Группа) _____ (Подпись, дата) Шемякин А. А. (И. О. Фамилия)

Преподаватель _____ (Подпись, дата) Строганов Ю. В. (И. О. Фамилия)

2023 г.

1 Теоретический раздел

1.1 Цель

1. Реализовать три метода построения вектора документа (словаря) на основе модели "мешок слов": в первом методе в качестве единицы анализа (ключа в словаре) используется лексема, во втором – словоформа, в третьем – часть речи. Оценкой единицы (значением в словаре) выступает частота её упоминания в документе.

2. Реализовать три меры близости: Жаккара, косинусную и евклидову.

3. Составить или найти в свободном доступе набор текстов (датасет), удовлетворяющий следующим требованиям:

   - каждый текст состоит не менее, чем из 3000 символов;
   - должно быть представлено не менее трёх текстов в каждом из пяти существующих стилей (научный, официально-деловой, публицистический, художественный, разговорный) по каждой из четырёх различных тематик (тематики выбираются исполнителем), таким образом получится минимум 3*5*4=60 текстов;
   - авторы текстов могут, но не обязаны быть разными.

4. Сформировать матрицы сравнения текстов из полученного набора данных с помощью каждой из трёх мер близости (Жаккара, косинусной и евклидовой) на основе каждого из трёх методов формирования вектора документа.

1.2 Меры близости

Мера Жаккара - это отношение мощности пересечения двух множеств к мощности их объединения. Мера Жаккара используется для измерения сходства между двумя множествами, и она может быть применена к набору слов или терминов, встречающихся в двух документах.

Формула меры Жаккара:

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|}$$

Косинусная мера - это мера близости, используемая для измерения угла между двумя векторами в многомерном пространстве. Она используется в векторном пространстве, чтобы измерить сходство между двумя векторами, представляющими два документа.

Формула косинусной меры:

$$\cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{|\mathbf{A}||\mathbf{B}|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}}$$

Евклидова мера - это мера близости, которая определяет расстояние между двумя точками в n-мерном пространстве. Евклидова мера используется для измерения сходства между двумя векторами, представляющими два документа.

Формула евклидовой меры:

$$d(\mathbf{A},\mathbf{B}) = \sqrt{\sum_{i=1}^{n}(A_i - B_i)^2}$$

1.3 Методы построения вектора документа

Опишем алгоритмы построения вектора документа (словаря) на основе модели "мешок слов": в первом методе в качестве единицы анализа (ключа в словаре) используется лексема, во втором – словоформа, в третьем – часть речи. Метод на основе лексем:

1. Документ разбивается на лексемы с помощью токенизации.

2. Для каждой лексемы в документе создается словарь, содержащий ее частоту в документе.

3. Вектор документа представляется как мешок слов, где каждый элемент вектора соответствует лексеме из словаря, а значение элемента - ее частота в документе.

Метод на основе словоформ:

1. Документ разбивается на слова с помощью токенизации.
2. Для каждой n-граммы в документе создается словарь, содержащий ее частоту в документе.
3. Вектор документа представляется как мешок слов, где каждый элемент вектора соответствует n-грамме из словаря, а значение элемента - ее частота в документе.

Метод на основе частей речи:

1. Документ разбивается на слова с помощью токенизации.
2. Каждое слово в документе присваивается часть речи с помощью POS-тэггера.
3. Для каждой части речи в документе создается словарь, содержащий ее частоту в документе.
4. Вектор документа представляется как мешок слов, где каждый элемент вектора соответствует части речи из словаря, а значение элемента - ее частота в документе.

2 Практический раздел

2.1 Составление датасета

Определим четыре тематики для текстов. Пусть это будет: экономика, наука, культура, политика. Для каждой тематики выберем по пять текстовых источников разных стилей. Сформируем датасет из следующих текстов. Тексты будем обрезать, чтобы количество символов было в пределах 3к-5к.

1. Экономика:

   (a) Научный стиль:

      i. Построение модели GVAR для российской экономики. Зубарев А.В., Кириллова М.А.

      ii. О получении стохастических прогнозов в детерминированной модели банковской системы России Радионов С.А.

      iii. Сравнение моделей прогноза волатильности криптовалют и фондового рынка Аганин А.Д., Маневич В.А., Пересецкий А.А., Погорелова П.В.

   (b) Официально-деловой стиль:

      i. Отчетность компаний ООО "ТИНЬКОФФ МОБАЙЛ"

      ii. Отчетность компаний ООО "1С МОБАЙЛ"

      iii. Отчетность компаний ООО "ФСИН"

   (c) Публицистический стиль:

      i. Бедность пересекла границу. Газета "Коммерсант"

      ii. Борис Титов предложил убрать из УК картельные сговоры на товарных рынках. Газета "Коммерсант"

      iii. Конкуренция становится более интеллектуальной. Газета "Коммерсант"

   (d) Художественный стиль:

      i. Книга "Богатый папа, бедный папа"Роберт Кийосаки

ii. Книга "Моя жизнь, мои достижения"Генри Форд

iii. Книга "Атлант расправил плечи"Айн Рэнд

(e) Разговорный стиль:

i. Сообщения с форума Финам.ру часть 1

ii. Сообщения с форума Финам.ру часть 2

iii. Сообщения с форума Финам.ру часть 3

2. Наука:

(a) Научный стиль:

i. Анализ особенностей расчета характеристик фонового излучения при решении задач лазерной локации в инфракрасном диапазоне спектра. Барышников Н.В., Степанов Р.О., Лебедев В.А.

ii. Влияние скорости ветра на точность сброса грузов с летательных аппаратов. Борейшо А.С., Савин А.В., Орлов А.Е., Гулевич С.П., Берг А.Г., Субботин В.Ю., Чернов В.Г., Евхаритский С.А., Герилович И.В.

iii. Разработка и валидация методики моделирования теплового и деформированного состояния деталей бесплатформенной инерциальной навигационной системы. Фролов А.В., Михайлов Ю.В., Смирнов С.В.

(b) Официально-деловой стиль:

i. Отчет лаборатории Международная лаборатория теории представлений и математической физики ВШЭ – Сколтех 2021

ii. Отчет лаборатории Международная лаборатория теории представлений и математической физики ВШЭ – Сколтех 2020

iii. Отчет лаборатории Международная лаборатория теории представлений и математической физики ВШЭ – Сколтех 2019

(c) Публицистический стиль:

i. Демо-номер журнала Наука и жизнь №3, 2023

ii. Журнал Квант №1, 2023

iii. Журнал Квант №1, 2022

(d) Художественный стиль:

i. Книга "Дюна"Герберт Фрэнк Патрик

ii. Книга "Черновик"Сергей Васильевич Лукьяненко

iii. Книга "Марсианские хроники"Рэй Дуглас Брэдбери

(e) Разговорный стиль:

i. Комментарии к постам N + 1 в группе Вконтакте часть 1

ii. Комментарии к постам Naked Science на их сайте

iii. Комментарии к постам N + 1 в группе Вконтакте часть 2

3. Культура:

(a) Научный стиль:

i. Архитектура в антропологическом измерении. Никифорова Л. В

ii. Картины Гюбера Робера в собрании канцлера князя А.А.Безбородко. Дерябина Е.В.

iii. Канон в архитектуре православного храма. Верховых Елена Юрьевна

(b) Официально-деловой стиль:

i. Отчеты о мероприятиях. МУК ДК "Соболевский"

ii. Отчеты о мероприятиях. МКУКМО "Среднеканская централизованная клубная система"

iii. Отчет о работе учреждений культуры Красносулинского района за 1-ое полугодие 2019 года

(c) Публицистический стиль:

i. На какие выставки стоит сходить в марте 2023-го. Журнал "Афиша"

ii. Главные сериалы весны-2023: про мафию, сантехников, рэперов и борьбу монашки с ChatGPT. Журнал "Афиша"

iii. Четырехдневная рабочая неделя: мировой опыт, исследования и ее (нескорое) будущее в России. Журнал "Афиша"

(d) Художественный стиль:

i. Книга "Луна и грош"Сомерсет Моэм

ii. Книга "Муки и радости"Ирвинг Стоун

iii. Книга "Воспоминания торговца картинами"Амбруаз Воллар

(e) Разговорный стиль:

i. Комментарии к постам Culture.ru в группе Вконтакте часть 1

ii. Комментарии к постам Culture.ru в группе Вконтакте часть 2

iii. Комментарии к постам Culture.ru в группе Вконтакте часть 3

4. Политика:

(a) Научный стиль:

i. "Государственный интерес"и гуманитарная дипломатия Оливера Кромвеля Л.И. Ивонина

ii. Русская небольшевистская дипломатия и вопрос международного признания Белого движения в 1918–1920 гг. Е. М. Миронова

iii. Гуманитарная помощь Красного Креста и других общественных организаций Нидерландов Советской России во время голода 1921–1923 гг. Г. Г. Циденков

(b) Официально-деловой стиль:

i. Аналитика. Выборы областных и городских парламентов: Ставки элит и вызовы для партий. Экспертный клюб "Регион".

ii. Аналитика. "Мы решаем не просто задачу импортозамещения, мы создаем промышленность будущего". Юрий Симачев

iii. Аналитика. От противостояния к созиданию: Сибирский поворот. Сергей Караганов

(c) Публицистический стиль:

i. Владимир Путин встретился с главой Чечни. Газета "Ведомости".

ii. В Думу внесли законопроект о повышении призывного возраста. Газета "Ведомости".

iii. В Белом доме началась активная фаза подготовки отчета премьер-министра перед Госдумой. Газета "Ведомости".

(d) Художественный стиль:

i. Книга "Капитал". Критика политической экономии Том 1. Карл Маркс

ii. Книга "1984". Оруэлл Джордж

iii. Книга "Воспоминания торговца картинами"Амбруаз Воллар

(e) Разговорный стиль:

i. Комментарии к постам ПОЛИТ.РУ в группе Вконтакте часть 1

ii. Комментарии к постам ПОЛИТ.РУ в группе Вконтакте часть 2

iii. Комментарии к постам ПОЛИТ.РУ в группе Вконтакте часть 3

2.2 Анализ

Чтобы определить, в каких случаях различные меры близости дают схожие результаты, а в каких - нет, можно визуализировать матрицы сходства с помощью тепловой карты.

Матрицы сравнения текстов из построенного набора данных с помощью каждой из трёх мер близости (Жаккара, косинусной и евклидовой) на основе каждого из трёх методов формирования вектора документа находятся в папке matrix.

Тепловые карты на основе этих матриц представлены на рис. 2.1

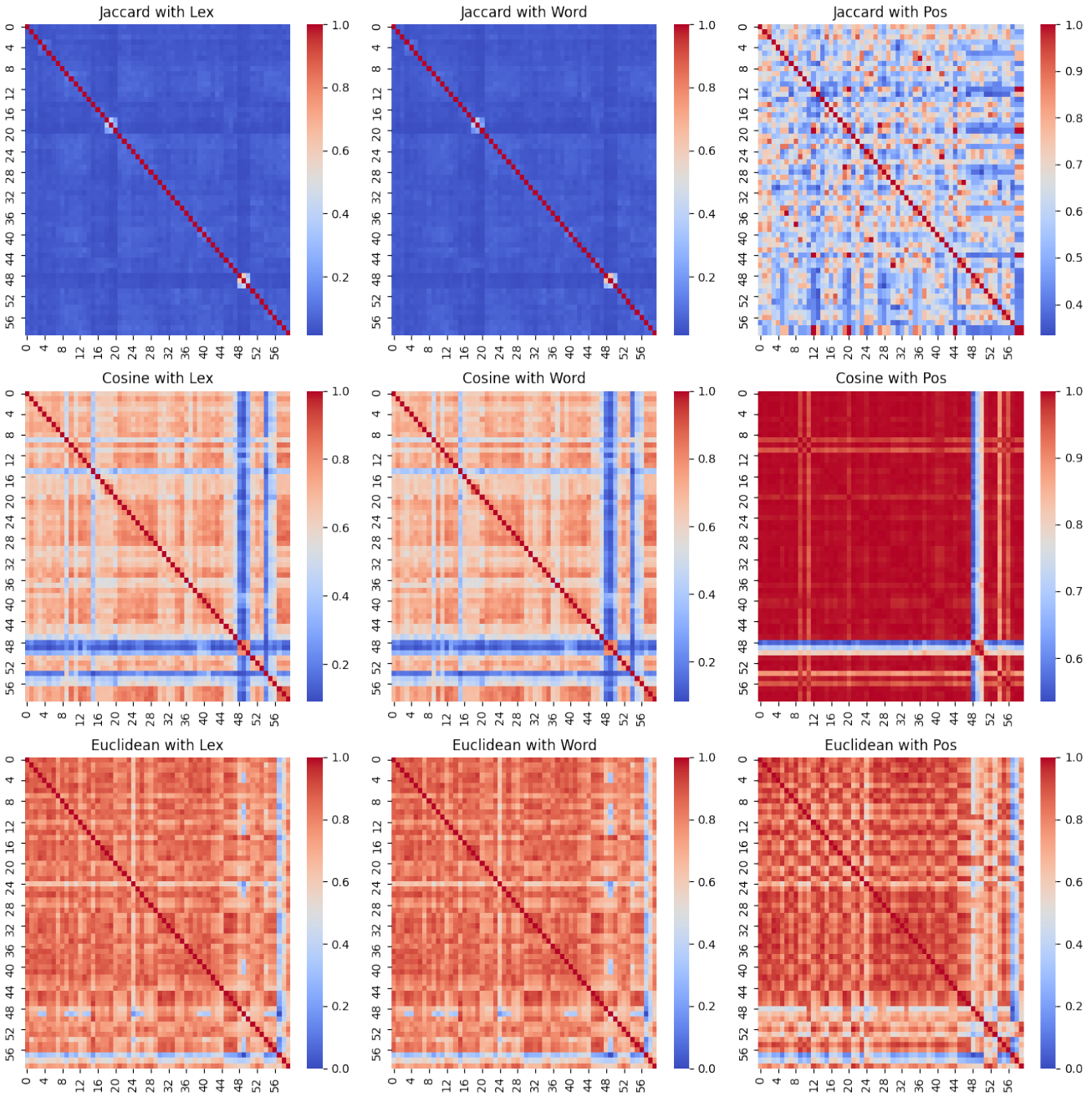

Рисунок 2.1 – Тепловая карта матриц близости

Тепловая карта средних значений близости этих матриц представлены на рис. 2.2

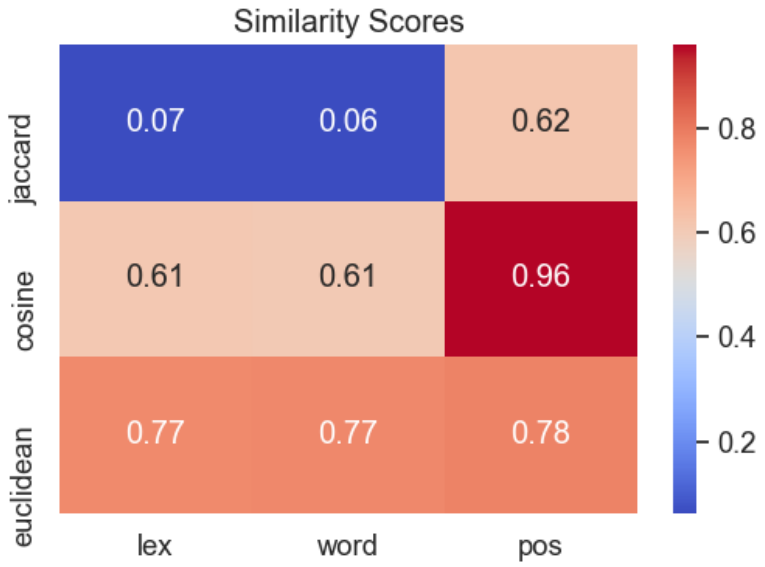

Рисунок 2.2 – Тепловая карта средних значений близости

Столбчата диаграмма средних значений близости этих матриц представлены на рис. 2.3

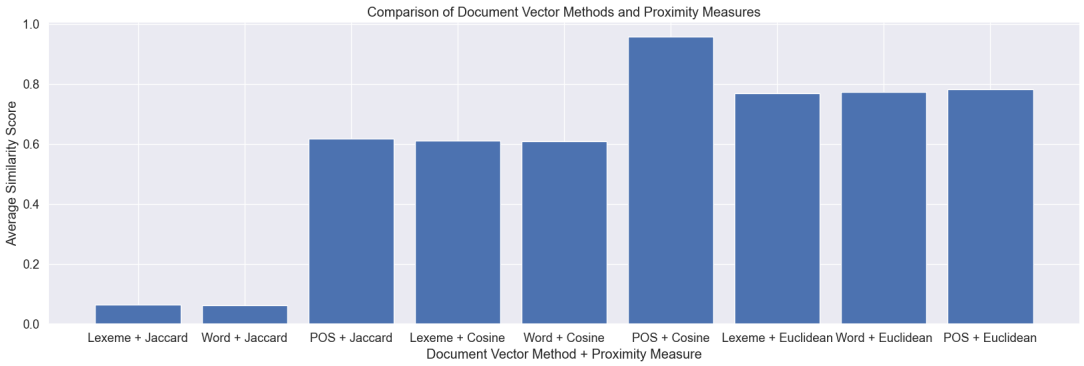

Рисунок 2.3 – Столбчатая диаграмма средних значений близости

Диаграмма зависимости между близостью текстов и стилем/тематикой представлена на рис. 2.4, где ось x представляет собой оценку близости между двумя текстами, а ось y - стиль или тематика текстов.

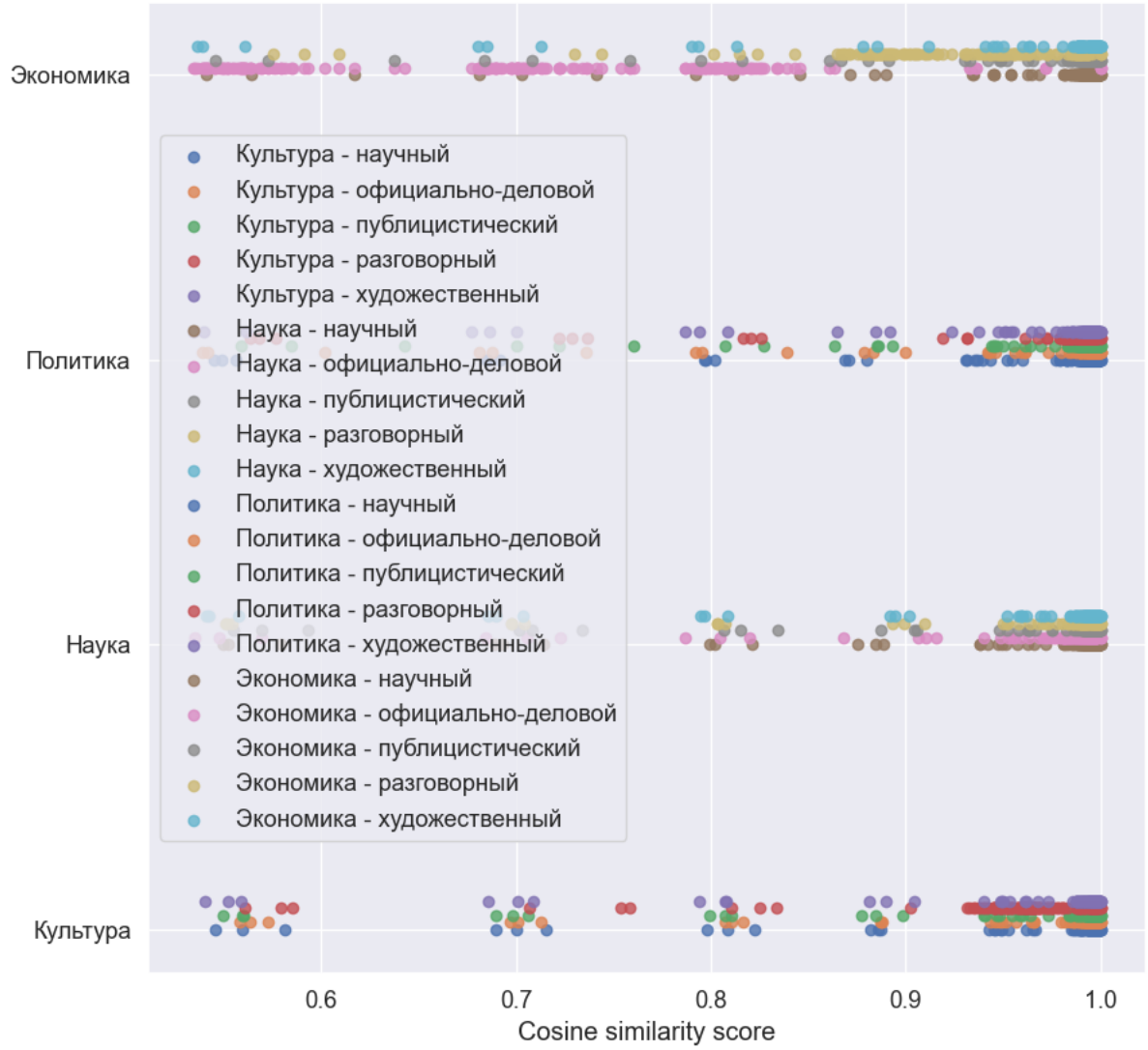

Рисунок 2.4 – Диаграмма рассеивания

Данные показывают, что мера близости Жаккара обычно дает низкие значения в сравнении с косинусной и евклидовой мерами близости, это указывает на то, она не является хорошей для этого набора данных.

Косинусная мера близости даёт более высокие значения, чем Жаккара, но более низкие, чем евклидова мера близости. Это указывает на то, что косинусная мера где-то улавливает аспекты сходства между документами, но не во всех случаях.

Евклидова мера близости даёт наиболее высокие значения по сравнению с двумя другими мерами, что указывает на то, что она может быть наилучшей мерой близости для этого набора данных.

Анализ комбинаций "метод векторного представления документа"и "меры близости"показал, что для меры близости Жаккара значения близости примерно одинаковы для всех трех методов векторного представления. Для косинусной и евклидовой меры близости значения сходства примерно одинаковы для методов, где ключами в словаре являются лексемы и словоформы, но значительно выше.

Таким образом, из этих результатов можно заключить, что наилучшей комбинацией "метод векторного представления документа"и "меры близости"для этого набора данных является "часть речи"и "косинусная поскольку она имеет наивысшее значение сходства 0.959.

3 Вывод

Реализованы три метода построения вектора документа и три меры близости. Составлен датасет. Сформированы матрицы сравнения текстов с помощью каждой из трёх мер близости на основе каждого из трёх методов формирования вектора документа. Следовательно, можно сделать вывод, что поставленная цель достигнута.

ПРИЛОЖЕНИЕ А

На листингах представлен исходный код программ на языке программирования Python.

Листинг А.1 – Исходный код программы

```
import nltk
nltk.download('punkt')
nltk.download('averaged_perceptron_tagger')
nltk.download('wordnet')

import pandas as pd
import numpy as np
from collections import defaultdict
import math
import matplotlib.pyplot as plt
import seaborn as sns

from nltk.stem import WordNetLemmatizer

df = pd.read_csv('dataset.csv')
df.head(15)

df['theme'].value_counts()

def build_dict_lex(text):
    tokens = nltk.word_tokenize(text, language = 'russian')
    freq_dict = defaultdict(int)
    for token in tokens:
        freq_dict[token.lower()] += 1
    return freq_dict

def build_dict_word(text):
    tokens = nltk.word_tokenize(text, language = 'russian')
    freq_dict = defaultdict(int)
    for token in tokens:
        freq_dict[token] += 1
    return freq_dict
```

```
33
34 def build_dict_pos(text):
35     tokens = nltk.word_tokenize(text, language="russian")
36     tagged_tokens = nltk.pos_tag(tokens)
37     freq_dict = defaultdict(int)
38     for token, pos in tagged_tokens:
39         freq_dict[pos] += 1
40     return freq_dict
41
42 def jaccard_similarity(v1, v2):
43     num = len(set(v1.keys()) & set(v2.keys()))
44     denom = len(set(v1.keys()) | set(v2.keys()))
45     return num / denom
46
47 def cosine_similarity(v1, v2):
48     dot_product = sum(v1[key] * v2.get(key, 0) for key in v1)
49     norm_v1 = math.sqrt(sum(v1[key]**2 for key in v1))
50     norm_v2 = math.sqrt(sum(v2[key]**2 for key in v2))
51     return dot_product / (norm_v1 * norm_v2)
52
53 def euclidean_distance(v1, v2):
54     common_keys = set(v1.keys()) & set(v2.keys())
55     return math.sqrt(sum((v1.get(k, 0) - v2.get(k, 0))**2 for k in
        common_keys))
56
57 lex_docs = []
58 word_docs = []
59 pos_docs = []
60 for i, row in df.iterrows():
61     text = row['text']
62     freq_dict_lex = build_dict_lex(text)
63     freq_dict_word = build_dict_word(text)
64     freq_dict_pos = build_dict_pos(text)
65     lex_docs.append(freq_dict_lex)
66     word_docs.append(freq_dict_word)
67     pos_docs.append(freq_dict_pos)
68
69 proximity_measures = ['jaccard', 'cosine', 'euclidean']
70 document_vector_methods = ['lex', 'word', 'pos']
```

```
for measure in proximity_measures:
    for method in document_vector_methods:
        similarity_matrix = np.zeros((len(df), len(df)))
        for i, vec1 in enumerate(eval(f'{method}_docs')):
            for j, vec2 in enumerate(eval(f'{method}_docs')):
                if i == j:
                    similarity_matrix[i, j] = 1
                    continue
                if measure == 'jaccard':
                    similarity = jaccard_similarity(vec1, vec2)
                elif measure == 'cosine':
                    similarity = cosine_similarity(vec1, vec2)
                elif measure == 'euclidean':
                    similarity = euclidean_distance(vec1, vec2)

                similarity_matrix[i, j] = similarity
        np.savetxt(f'matrix/{measure}_{method}.csv',
           similarity_matrix, delimiter=',')

jl = pd.read_csv('matrix/cosine_lex.csv')
jl.head()



fig, axs = plt.subplots(len(proximity_measures),
   len(document_vector_methods), figsize=(13, 13))

for i, measure in enumerate(proximity_measures):
    for j, method in enumerate(document_vector_methods):
        similarity_matrix =
           np.genfromtxt(f'matrix/{measure}_{method}.csv',
           delimiter=',')

        if measure == 'euclidean':
            max_distance = np.max(similarity_matrix)
            similarity_matrix = 1 - (similarity_matrix /
               max_distance)

        for k in range(len(similarity_matrix)):
```

```
105                 similarity_matrix[k, k] = 1
106
107             ax = sns.heatmap(similarity_matrix, ax=axs[i, j],
                    cmap='coolwarm')
108             ax.set_title(f'{measure.capitalize()} with
                    {method.capitalize()}')
109
110 plt.tight_layout()
111 plt.savefig('images/heatmap.png', dpi = 100 )
112
113
114 similarity_matrix = np.zeros((len(proximity_measures),
        len(document_vector_methods)))
115
116 for i, measure in enumerate(proximity_measures):
117     for j, method in enumerate(document_vector_methods):
118         data = np.genfromtxt(f'matrix/{measure}_{method}.csv',
                delimiter=',')
119         if measure == 'euclidean':
120             max_distance = np.max(data)
121             data = 1 - (data / max_distance)
122         mean_similarity = np.mean(data)
123         similarity_matrix[i, j] = mean_similarity
124
125 sns.set(font_scale=1.2)
126 ax = sns.heatmap(similarity_matrix, cmap='coolwarm', annot=True,
        fmt='.2f',
127                  xticklabels=document_vector_methods,
                        yticklabels=proximity_measures)
128 ax.set_title('Similarity Scores')
129 plt.savefig('images/similarity_scores.png', dpi = 100 )
130
131
132 jaccard_lex = np.loadtxt('matrix/jaccard_lex.csv', delimiter=',')
133 jaccard_word = np.loadtxt('matrix/jaccard_word.csv', delimiter=',')
134 jaccard_pos = np.loadtxt('matrix/jaccard_pos.csv', delimiter=',')
135 cosine_lex = np.loadtxt('matrix/cosine_lex.csv', delimiter=',')
136 cosine_word = np.loadtxt('matrix/cosine_word.csv', delimiter=',')
137 cosine_pos = np.loadtxt('matrix/cosine_pos.csv', delimiter=',')
```

```
euclidean_lex = np.loadtxt('matrix/euclidean_lex.csv',
   delimiter=',')
euclidean_word = np.loadtxt('matrix/euclidean_word.csv',
   delimiter=',')
euclidean_pos = np.loadtxt('matrix/euclidean_pos.csv',
   delimiter=',')

jaccard_lex_avg = np.mean(jaccard_lex)
jaccard_word_avg = np.mean(jaccard_word)
jaccard_pos_avg = np.mean(jaccard_pos)

cosine_lex_avg = np.mean(cosine_lex)
cosine_word_avg = np.mean(cosine_word)
cosine_pos_avg = np.mean(cosine_pos)

max_distance = np.max(euclidean_lex)
data = 1 - (euclidean_lex / max_distance)
euclidean_lex_avg = np.mean(data)

max_distance = np.max(euclidean_word)
data = 1 - (euclidean_word / max_distance)
euclidean_word_avg = np.mean(data)

max_distance = np.max(euclidean_pos)
data = 1 - (euclidean_pos / max_distance)
euclidean_pos_avg = np.mean(data)

results = pd.DataFrame({
    'Document Vector Method': ['Lexeme', 'Word', 'POS']*3,
    'Proximity Measure': ['Jaccard']*3 + ['Cosine']*3 +
       ['Euclidean']*3,
    'Average Similarity Score': [
        jaccard_lex_avg, jaccard_word_avg, jaccard_pos_avg,
        cosine_lex_avg, cosine_word_avg, cosine_pos_avg,
        euclidean_lex_avg, euclidean_word_avg, euclidean_pos_avg
    ]
})

plt.figure(figsize=(20, 6))
```

```
plt.bar(results.index, results['Average Similarity Score'])
plt.xticks(results.index, results['Document Vector Method'] + ' + ' + results['Proximity Measure'])
plt.xlabel('Document Vector Method + Proximity Measure')
plt.ylabel('Average Similarity Score')
plt.title('Comparison of Document Vector Methods and Proximity Measures')
plt.savefig('images/comparison.png', dpi = 100 )

results

measure = 'cosine'
method = 'pos'

similarity_matrix = np.genfromtxt(f'matrix/{measure}_{method}.csv', delimiter=',')
if measure == 'euclidean':
    max_distance = np.max(similarity_matrix)
    similarity_matrix = 1 - (similarity_matrix / max_distance)

theme_labels = df['theme'].unique()
style_labels = df['style'].unique()

fig, ax = plt.subplots(figsize=(10, 10))
for i, theme in enumerate(theme_labels):
    theme_mask = df['theme'] == theme
    for j, style in enumerate(style_labels):
        style_mask = df['style'] == style
        mask = theme_mask & style_mask
        similarity_scores = similarity_matrix[mask].flatten()
        ax.scatter(similarity_scores,
            [i+j*0.025]*len(similarity_scores), label=f'{theme} - {style}', alpha=0.8)

ax.set_yticks(range(len(theme_labels)))
ax.set_yticklabels(theme_labels)
ax.set_xlabel(f'{measure.capitalize()} similarity score')
legend = ax.legend(bbox_to_anchor=(0, 1), loc='lower center')
plt.legend()
```

```
207 fig.savefig('images/relationship.png', dpi=100)
```

Листинг А.2 – Исходный код программы

```
import os
import pandas as pd

# Создаем пустой датасет
dataset = pd.DataFrame(columns=['text', 'style', 'theme'])

# Путь к папке с текстами
path = 'data'

# Список стилей и тематик
styles = ['научный', 'официально-деловой', 'публицистический',
    'художественный', 'разговорный']
themes = ['Наука', 'Экономика', 'Культура', 'Политика']
# Обход всех файлов в папке
for root, dirs, files in os.walk(path):
    for file in files:
        # Чтение файла
        with open(os.path.join(root, file), 'r', encoding='utf-8')
            as f:
            text = f.read()

        # Извлечение стиля и тематики из имени файла
        theme, style, _ = file.split('_')

        # Добавление текста в датасет
        dataset = dataset.append({'text': text, 'style': style,
            'theme': theme}, ignore_index=True)

# Сохранение датасета в CSV файл
dataset.to_csv('dataset.csv', index=False)
```