М. С. Воробьева, А. А. Дубаков

Тюменский государственный университет, г. Тюмень

УДК 004.912

РАЗРАБОТКА ПРИЛОЖЕНИЯ ДЛЯ ОПРЕДЕЛЕНИЯ ТЕМАТИКИ ТЕКСТА С ИСПОЛЬЗОВАНИЕМ АЛГОРИТМОВ КЛАСТЕРИЗАЦИИ

Аннотация. В статье рассмотрена задача кластеризации студенческих квалификационных работ по ключевым словам. Разработано приложение для вывода результатов кластеризации. Рассмотрены подходы к визуализации многомерных данных.

Ключевые слова. Интеллектуальный анализ текстовых данных, выделение ключевых слов, кластеризация, многомерные данные, визуализация.

Введение

Обширной областью работы с текстами является интеллектуальный анализ, целью которого является получение информации из коллекций текстовых документов, основываясь на применении эффективных методов машинного обучения и обработки естественного языка.

В рамках исследования рассматривается задача кластеризации текстовых документов. Данная статья основывается на работе [6], где показано, как в приложении «Clustering Analysis» с помощью кластерного анализа текстовых документов строятся кластеры, выводятся результаты в виде дендрограммы, что позволяет определить структуру корпуса документов.

Для проведения анализа были выбраны квалификационные работы студентов института математики и компьютерных наук, в которых нужно определить основные направления исследования, тематики работ и выполнить кластеризацию всех документов.

Актуальность работы заключается в том, что полученные данные позволяют выявить наиболее распространенные тематики, встречающиеся

в студенческих работах, а значит, определить, какие дисциплины наиболее популярны у студентов.

Целью работы является разработка приложения для интеллектуального анализа текстовых работ студентов.

Для достижения цели были поставлены следующие задачи:

- изучение технологий для кластеризации текстовых данных и подходов к визуализации результатов;
- разработка программного модуля для кластеризации текстовых документов;
- разработка приложения для визуализации результатов обработки документов и анализа направлений обучения.

Технологии интеллектуального анализа текстовых данных

Существуют несколько основных подходов к выделению ключевых слов из текстовых документов [1]:

- «мешок слов» (анг. Bag of Words): текст представляется в виде набора слов, составленного без учета грамматики и порядка следования слов в тексте. Для каждого отдельного слова вычисляется коэффициент значимости;
- тематическое моделирование, когда для коллекции текстовых документов строится тематическая модель, которая определяет, к каким темам относится каждый из документов. Тематическая модель (англ. topic model) коллекции текстовых документов определяет, к каким темам относится каждый документ и какие слова (термины) образуют каждую тему;
- семантические модели (например, word2vec): значимость слов определяется исходя из взаимного расположения всех слов документа внутри семантической модели.

С целью подробного изучения кластерного анализа текстовых данных было разработано пользовательское приложение «Clustering Analysis» на языке программирования C# с использованием технологии WPF

В рамках данной работы выбран первый подход. Представив документ как набор слов и вычислив коэффициент значимости для каждого уникального слова, можно составить вектор документа, отражающий степень присутствия в тексте того или иного понятия. Если пересечь такие вектора всех документов в корпусе, то образуется матрица коэффициентов значимости, и все тексты окажутся в одном векторном пространстве. Так как в дальнейшем планируется кластеризация документов, этот фактор является ключевым при выборе данного подхода.

Для расчета коэффициента значимости выбрана метрика tf-idf, при вычислении которой учитывается частота встречаемости слова во всем наборе документов, что позволяет отфильтровать незначимые слова.

Для кластеризации выбран алгоритм k-средних. Основная идея заключается в том, что на каждой итерации вычисляется центр масс для каждого кластера, полученного на предыдущем шаге, затем векторы разбиваются на кластеры вновь в соответствии с тем, какой из новых центров оказался ближе по выбранной метрике. Алгоритм завершается, когда на очередной итерации не происходит изменения расстояния между кластерами. Это происходит за конечное число итераций, так как количество возможных разбиений конечного множества конечно, а на каждом шаге суммарное квадратичное отклонение уменьшается, поэтому зацикливание невозможно.

Результаты работы алгоритма k-средних позволяют выявить взаимное расположение текстов в векторном пространстве, что облегчает визуализацию кластерной структуры корпуса документов. Недостатком алгоритма является необходимость заранее задавать количество кластеров. Для решения этой проблемы используется метод силуэтов [2].

Подготовка исходных данных

Извлечение текстовых данных из исходных файлов в формате pdf производится с помощью программы, написанной на языке C#, для парсинга разметки pdf используется библиотека iTextSharp. После извлечения текста происходит дополнительная обработка файлов: фильтруются небуквенные

символы, в том числе знаки препинания и табуляции, все буквы слов переводятся в нижний регистр.

Разработка модуля для обработки данных

Программная реализация обработки текстовых данных выполнена на языке Python [3]. В рамках работы были использованы библиотеки:

- sklearn – для вычисления метрики tf-idf, кластеризации и понижения размерности векторов
- nltk – для подготовки, стемминга и токенизации текста
- numpy – для хранения больших массивов вещественных чисел
- pandas – для организации хранения и вывода в файл данных о текстовых документах, включая результаты анализа

Обработка текстовых документов происходит следующим образом:

- стемминг и токенизация документов, фильтрация знаков препинания

и стоп-слов

- подсчет значимости слов относительно всего набора документов с помощью метрики tf-idf
- составление векторов документов из значений tf-idf по глобальному для всего набора документов словарю ключевых слов, в результате получается матрица размерностью N на M, где N – количество документов, M – количество уникальных токенов во всем наборе документов
- кластеризация текстовых документов используя вектора
- преобразование матрицы, состоящей из векторов документов, в набор двумерных векторов с помощью алгоритма LDA для дальнейшей визуализации [4].

Блок-схема алгоритма обработки данных представлена на рисунке 1.

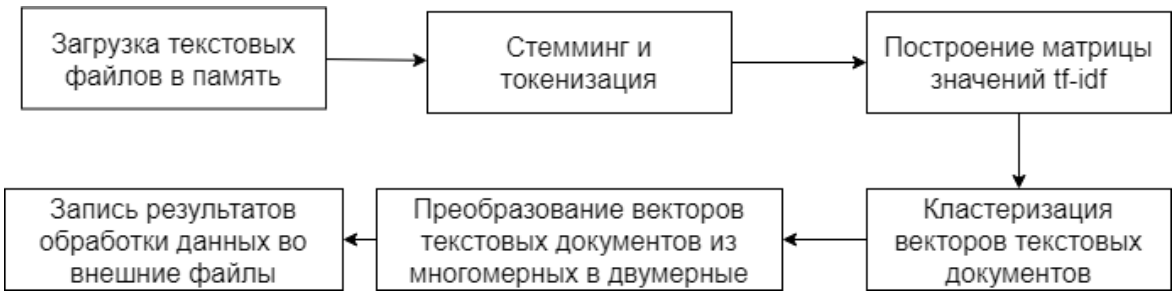

Рис. 1. Схема работы модуля обработки данных.

На каждом из этапов работы программы записываются результаты обработки данных, а именно:

- результаты токенизации – глобальный для всего набора документов словарь, в котором каждому токену соответствует полная форма этого токена (слово без применения стемминга);
- матрица tf-idf, где каждому документу соответствует вектор, столбцами которого являются значения tf-idf для каждого из слов в глобальном словаре;
- массив, содержащий информацию о номере кластера, которому принадлежит каждый из текстовых документов;
- массив точек, представляющих положение документов на двумерной плоскости, после преобразования матрицы tf-idf с помощью алгоритма LDA.

Таким образом, благодаря возможности загрузки сохраненных данных каждый шаг может быть выполнен без запуска предыдущих этапов, основное назначение файлов — передача данных в визуализатор.

За стемминг, токенизацию и построение матрицы значений tf-idf отвечает класс spaVectorizer (см. табл. 1). За кластеризацию и многомерное преобразование матрицы значений tf-idf отвечает класс spaClusterizer.

Таблица 1. Методы класса spaVectorizer.

Название метода	Возвращаемое значение	Описание
get_text_from_file	Массив содержимого файлов и названий файлов	Загружает содержимое файлов в память программы
tokenize_and_stem	Массив строковых токенов	Производит стемминг и фильтрацию лишних символов для каждого слова в текстах
tokenize_only	Массив строковых токенов	Производит только фильтрацию лишних символов для каждого слова в текстах
vectorize_text_tfidf	Матрица значений tf-idf	Производит векторизацию текстов с помощью метрики tf-idf

Разработка приложения для визуализации

Программный модуль для обработки данных предоставляет первичные результаты интеллектуального анализа документов. Для дальнейшего изучения полученной информации потребовалось разработать приложение для визуализации данных.

Визуализатор является пользовательским приложением для ОС Windows, написанном на языке C# с использованием технологии WPF [5]. Основное назначение приложения – предоставить пользователю визуальное представление результатов кластерного анализа и упростить их интерпретацию.

Приложение обладает следующим функционалом:

- вывод на экран результаты кластеризации в виде набора точек на плоскости, где каждая точка представляет собой документ (см. рис. 2)
- вывод на экран список документов и список кластеров (см. рис. 2)
- при выборе документа на плоскости (при нажатии на точку) или в списке, на экран выводится детальная информация о выбранном объекте

- при выборе кластера в списке кластеров, открывается новое окно и входящие в кластер документы отображаются на плоскости так же, как и в основном окне, но вместо списка кластеров выводится детальная информация о выбранном кластере (см. рис. 3).

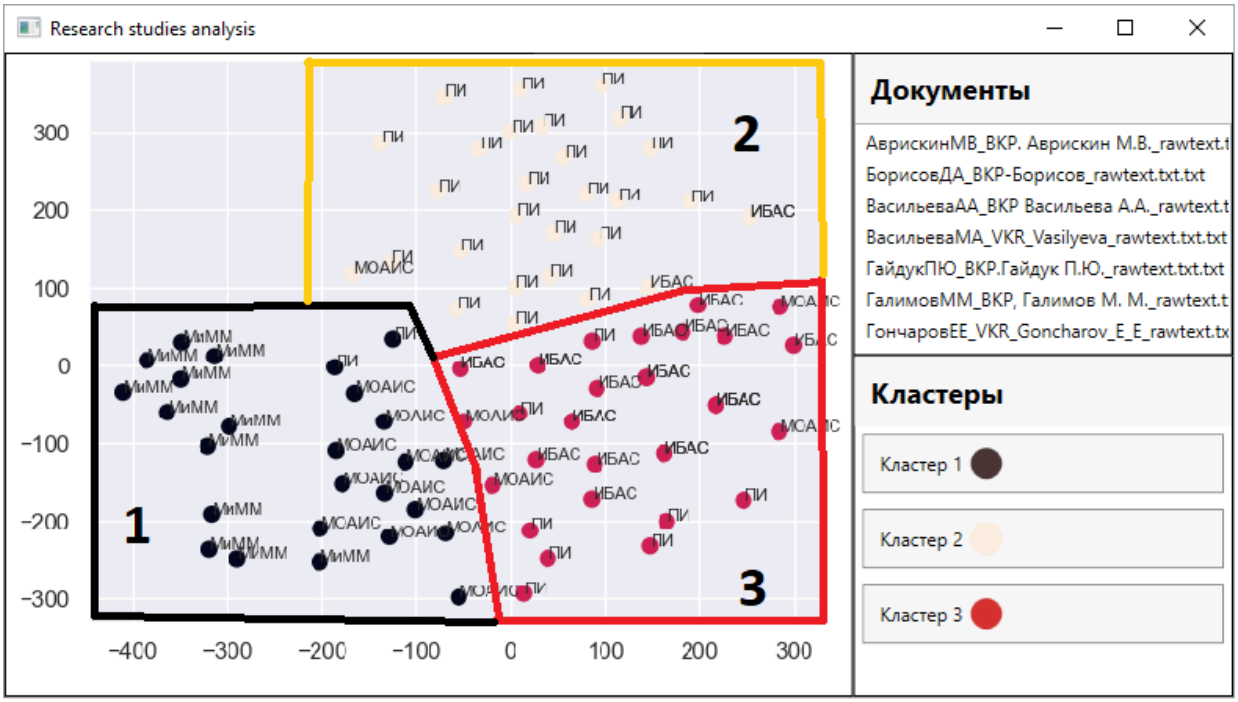

Рис. 2. Главное окно программы.
Область 1 — кластер 1, область 2 — кластер 2, область 3 — кластер 3

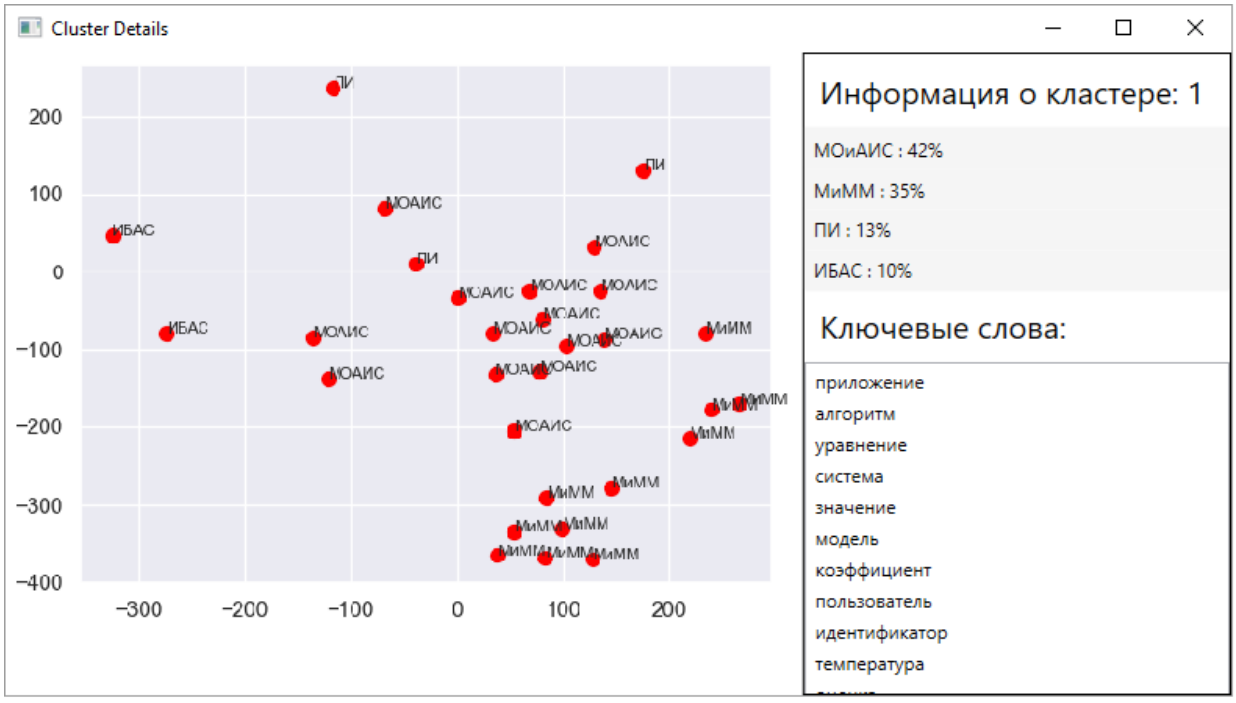

Рис. 3. Детальная информация о кластере 1.

Перед началом работы программы данные загружаются из указанных файлов, в приложении написана система классов для хранения файлов (см. табл. 2).

Таблица 2. Система классов приложения

Класс	Назначение
GlobalData	общая информация о данных: словарь ключевых слов корпуса документов, номера кластеров для каждого документа, набор точек для визуального представления кластерной структуры и набор классов, представляющих каждый отдельный документ
DocumentData	информация о документе: название, автора, год написания, номер кластера, вектор значений tf-idf
VisualData	информация о документе, необходимая для визуализации (координаты точки на основной диаграмме, цвет точки и подпись для вывода на экран, словарь, состоящий из ненулевых значений вектора tf-idf и соответствующих им словам из глобального словаря, а также связанный экземпляр класса DocumentData)

Для демонстрации работы программы были обработаны файлы с дипломными работами за 2018 год по направлениям обучения: Математическое обеспечение и администрирование информационных систем (МОАИС), Механика и математическое моделирование (МиММ), Информационная безопасность автоматизированных систем (ИБАС), Прикладная информатика (ПИ).

Кластерная структура набора данных соответствует 3 кластерам.

- Кластер 1 содержит работы по направлениям МиММ, МОАИС, в меньшей степени ПИ и ИБАС. Общими ключевыми словами являются:

«алгоритм», «уравнение», «приложение», «точка», «значение», «коэффициент», «параметр» и др.

- В кластере 2 располагаются работы по направлению ПИ и несколько работ МОАИС, МиММ и ИБАС. Ключевые слова кластера: «клиент», «система», «пользователь», «заявка», «сервер», «электронный ресурс» и др.
- В кластере 3 сосредоточены в основном работы по направлениям МОАИС, ИБАС и ПИ. Ключевыми словами являются «информационная система», «доступ», «угроза», «сервер», «база данных», «персональные данные» и др.

Информация о соотношении направлений обучения в кластерах представлена в таблице 3.

Таблица 3. Соотношение направлений обучения в кластерах

Направление	Кластер 1		Кластер 2		Кластер 3	
	Кол-во	Соотношение, %	Кол-во	Соотношение, %	Кол-во	Соотношение, %
МОАИС	13	41,94	1	3,45	6	22,22
МиММ	11	35,48	1	3,45	0	0
ИБАС	3	9,68	4	13,79	13	48,15
ПИ	4	12,9	23	79,31	8	29,63

Проанализировав кластерную структуру набора документов, можно сделать вывод о том, что тематика работ студентов соответствует особенностям направлений обучения. МОАИС находится на стыке двух основных тематик – работы студентов этого направления находятся как в кластере с МиММ (что говорит о математической подготовке), так и в кластере с ИБАС (что соответствует тематике администрирования информационных систем). Около 80% работ по направлению ПИ находятся в отдельном кластере.

При рассмотрении кластерной структуры на рисунке 2 можно отметить, что работы по каждому из направлений обучения чаще всего соседствуют друг с другом. Другими словами, визуально легко выделить кластера, каждый из которых соответствовал бы определённому направлению обучения, что свидетельствует о сходстве по тематике работ обучающихся.

Заключение

В результате были изучены технологии интеллектуального анализа текстов, подходы к визуализации результатов текстового анализа, рассмотрена задача выделения ключевых слов из текстовых документов, изучены особенности кластеризации многомерных данных. Разработан модуль на языке Python для извлечения ключевых слов и кластерного анализа текстовых данных.

На языке C# с использованием технологии WPF разработано приложение для визуализации результатов анализа текста, которое позволяет визуально оценивать кластерную структуру набора документов и предоставляет данные для анализа причин формирования кластеров.

В ходе эксперимента было выявлено, что чаще всего работы одного направления либо находятся в одном кластере, либо принадлежат нескольким соседним кластерам.

СПИСОК ЛИТЕРАТУРЫ

1. Технологии анализа данных: Data Mining, Visual Mining, Text Mining, OLAP / И.И. Холод, В.В. Степаненко, М.С. Куприянов. – М.: БХВ-Петербург, 2007. – 384 с.
2. Мьятт, Г. Making Sense of Data I: A Practical Guide to Exploratory Data Analysis and Data Mining / Г. Мьятт, В. Джонсон. – М.: Wiley, 2-е изд. 2014. – 248 с.
3. Маккини, У. Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython / У. Маккини. – М.: O’Reilly Media, 2-е изд., 2017. – 550 с.

4. Reduction – Zenwa | Python Machine Leatning Tutorials https:// pythonmachinelearning.pro/dimensionality-reduction/ (Дата последнего обращения: 28.05.19)
6. Макдональд, М. Pro WPF 4.5 in C#: Windows Presentation Foundation in .NET 4.5 / М. Макдональд. – М.: Apress, 4-е изд., 2012. – 1078 с.
7. Дубаков, А.А., Воробьев, А.М. Разработка алгоритма иерархической агломеративной кластеризации для анализа текстовых документов // Математическое и информационное моделирование: сборник научных трудов. Вып. 16. – Тюмень: Издательство Тюменского государственного университета, 2018. – 534 с.