Министерство образования и науки Российской Федерации
Федеральное государственное автономное образовательное учреждение высшего образования
«САНКТ-ПЕТЕРБУРГСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ПЕТРА ВЕЛИКОГО»

Институт компьютерных наук и технологий
Кафедра Компьютерные интеллектуальные технологии

«Допустить к защите»
Зав. каф. КИТ, к.т.н.
_____ А.В. Речинский
«____» _____ 20___ г.

МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

ИССЛЕДОВАНИЕ МЕТОДИК СЕМАНТИЧЕСКОГО АНАЛИЗА ТЕКСТОВ В ЗАДАЧАХ КЛАСТЕРИЗАЦИИ ТЕКСТОВЫХ СООБЩЕНИЙ

направление: 02.04.03 «Математическое обеспечение и администрирование информационных систем»

Выполнил(а):
Баудин Илья Дмитриевич
Подпись_____

Руководитель:
доцент каф. КИТ ИКНТ, к.т.н.,
Щукин Александр Валентинович
Подпись_____

Консультант:
ст. преподаватель каф. КИТ ИКНТ,
Туральчук Константин Анатольевич
Подпись_____

Рецензент:
директор ВИШ ИДО,
Кудаков Александр Владимирович
Подпись_____

Санкт-Петербург
2017

РЕФЕРАТ

«Исследование методик семантического анализа текстов в задачах кластеризации текстовых сообщений»

Работа содержит: стр. 72, ил. 21, табл. 5, библ.: 27.

Ключевые слова: обработка естественного языка, компьютерная лингвистика, извлечение информации, кластеризация, кластеризация текстовых сообщений, Томита-парсер, GLR-анализатор.

Кластеризация документов применяется уже достаточно долгое время, начиная с времен бурного роста электронно-вычислительных машин. Но сама по себе задача кластеризации относится к разделам data mining. В данной работы предлагается методика улучшения качества кластеризации текстовых сообщений на естественном языке, планируется увеличить точность и качество кластеризации путем учета семантической информации исходных текстов. Проводимое исследование включает в себя обзор имеющихся методов кластеризации текстовой информации, анализ методов извлечения именованных сущностей из текстов на естественном языке, разработку и реализацию методики кластеризации текстовых сообщений на русском языке с учетом семантической информации.

«Research methods for semantic analysis of texts in text clustering»

This thesis includes: pages, figures, tables, references.

Keywords: natural language processing, computional linguistic, information extraction, clustering, text clustering, Tomita-parser, GLR-parser.

Clustering documents has been used for quite a long time, since the times of rapid growth of electronic computers. But the clustering task itself refers to the data mining sections. In this paper, we propose a technique for improving the quality of text message text clustering in a natural language, it is planned to increase the accuracy and quality of clustering by taking into account the semantic information of source texts. The research includes a review of available methods for clustering textual information, analysis of methods for extracting named entities from texts in natural language, development and

implementation of a method for clustering text messages in Russian with regard to semantic information

ОГЛАВЛЕНИЕ

ОПРЕДЕЛЕНИЯ

В выпускной квалификационной работе используются следующие термины с соответствующими определениями:

- машинное обучение – это раздел искусственного интеллекта, использующий разделы математической статистики, численных методов оптимизации, теории вероятностей, дискретного анализа для извлечения знаний из данных;
- грамматика – множество правил на языке КС-грамматик, описывающих синтаксическую структуру выделяемых цепочек;
- терминальный символ – объект формальной грамматики, имеющий в нём конкретное неизменяемое значение и являющийся элементом построения слов данного языка;
- нетерминальный символ (нетерминал) – объект, обозначающий какую-либо сущность языка у которой отсутствует конкретное символьное значение.
- кластеризация - задача разбиения множества объектов на группы, называемые кластерами. Внутри каждой группы должны оказаться «похожие» объекты, а объекты разных группы должны быть как можно более отличны;
- data mining – собирательное название, используемое для обозначения совокупности методов обнаружения в данных ранее неизвестных, нетривиальных, практически полезных и доступных интерпретации знаний, необходимых для принятия решений в различных сферах человеческой деятельности;
- тезаурус – в современной лингвистике, особая разновидность словарей, в которых указаны семантические отношения между лексическими единицами;
- неоднозначная грамматика (неоднозначность) – это грамматика, написанная на формальном языке, которая может породить результат более чем одним способом;
- семантическая информация – смысловой аспект, отражающий отношение между формой сообщения и его смысловым содержанием;

- факт – таблицы с колонками, которые называются полями фактов. Факты заполняются во время анализа парсером предложения;
- помета – конструкции языка, используемые для наложения ограничений на терминал или нетерминал.

СПИСОК ОБОЗНАЧЕНИЙ И СОКРАЩЕНИЙ

NLP – Natural Language Processing

EM – Expectation Maximization

IE – Information Extraction

MDL – Minimun Description Length

LSA – Latent Semantic Analysis

pLSA – Probabilistic Latent Semantic Analysis

ML – Machine Learning

API – Application Programming Interface

GLR – Generalized Left-to-rignt Rightmost

JAPE – Java Annotation Patterns Engine

AGFL – Affix Grammars over a Finite Lattice

LSPL – Lexico-Syntactic Pattern Language

GNU GPL – GNU General Public License

TXT – Text file

SMM – Social Media Marketing

СКО – Среднеквадратическое отклонение

ФИО – фамилия, имя, отчество

ВВЕДЕНИЕ

В современном мире происходит бурное развитие информационных технологий, в связи с чем, растут объемы цифровой информации, в том числе и информации, представленной естественно-языковыми средствами. Все большую актуальность приобретают вопросы, связанные с ее обработкой и дальнейшим анализом. Привычный для нас, естественный язык является основным способом представления информации и обладает рядом признаков, которые затрудняют его понимание вычислительной техникой. К таким признакам можно отнести неоднозначность терминов, избыточность, конвенциональность и непрозрачность.

Задача распознавания текстов на естественном языке может решаться с использованием различных технологий и методов, в первую очередь на базе методов обработки данных, представленных естественно-языковыми средствами - NLP (Natural Language Processing). Кластеризация текстовых документов, то есть разбиение документов на определенное, заранее не известное количество подмножеств, помеченных какими-то семантическими описателями одна из приоритетных задач, решаемых различными информационными системами. [1]

Целью выпускной квалификационной работы является исследование и разработка методики семантического анализа текстов в задачах кластеризации текстовых сообщений.

В соответствии с поставленной целью необходимо решить следующие задачи:

- анализ существующих методов кластеризации текстовых документов;
- анализ существующих методов семантического анализа текстов;
- построение методики формирования кластеров с использованием семантической информации;
- исследование методики на примере кластеризации текстов.

Кластеризация документов применяется уже достаточно долгое время, начиная с времен бурного роста электронно-вычислительных машин. Но сама по себе задача кластеризации относится к разделам data mining. В результате данной

работы планируется увеличить точность и качество кластеризации путем учета семантической информации исходных текстов. Информации о применении данных подходов ранее обнаружено не было.

Объектом исследования является задача кластеризации текстов и способы улучшения качества получаемых результатов.

Теоретической основой данного исследования служили публикации в иностранных изданиях [2], блоги и инструменты таких IT компаний, как Яндекс, IBM, Google, Открытые системы, АОТ.

Практическая значимость исследования заключается в результатах сравнения существующих методик кластеризации текстовых документов и новой методики кластеризации, учитывающей семантическую информацию исходных текстов.

Выпускная квалификационная работа состоит из реферата, введения, четырех глав, заключения, списка используемых источников, приложений.

В первой главе работы рассматриваются основные определения, касающиеся задачи кластеризации и существующих проблем в данной области, а также анализируются существующие методы кластеризации и их применение для текстовых данных.

Во второй главе рассматривается теоретическая основа данной работы, предлагается алгоритм обработки данных и рассматриваются инструменты, с помощью которых, данный алгоритм будет реализован, вводятся метрики, по которым можно будет посчитать близость признаков в задаче кластеризации и оценить полученные результаты.

В третьей главе данной работы определяются инструменты для практической реализация каждого модуля системы, предложенная методика реализуется с помощью рассмотренных средств, объясняется назначение и принцип работы основных компонентов системы.

В четвертой главе проводится тестирование разработанной системы, и сравниваются результаты с аналогичными алгоритмами выполнения

кластеризации, приводятся визуальные результаты работы алгоритма и формально оцениваются полученные результаты с помощью численных оценок.

1 Задача кластеризации текстовых документов

1.1 Основные определения

Одной из самых распространенных форм представления знаний являются тексты на естественном языке. Текстовая информация естественная для человека, она легко создается, воспринимается, распространяется и изменяется. Количество информационных ресурсов в современном мире неуклонно растет из-за новых возможностей позиционирования текстов, связанных с масштабным развитием электронно-вычислительной техники, а также с повышением доступности методов записи и хранения информации. Необходимость изучения и осмысления постоянно растущего объема неструктурированной текстовой информации делает задачу анализа этих данных актуальной на сегодняшний день. [3]

Задача кластеризации данных – задача по объединению в группы объектов, схожих по определенным признакам – один из фундаментальных вопросов Data Mining. Зачастую, кластеризация данных имеет прикладной характер и список областей, в которых применяется кластеризация достаточно широк: анализ текстов, сегментация изображений, прогнозирование различных событий, маркетинг. В современных задачах кластеризация является первым этапом обработки данных для формирования групп признаков, для которых, в дальнейшем будут применены другие методы и модели.

Стоит отметить, что задача кластеризации документов имеет много общего с задачей классификации текстов в заранее созданную и предварительно заполненную систему категорий. Несмотря на предварительное сходство, кластеризация имеет ряд своих особенностей, которые необходимо учитывать при решении задач. Вопреки хорошо изученным и эффективным на практике методам классификации, подходы к решению задачи кластеризации в некоторой степени бедны и имеют весьма ограниченную практическую применимость. Основная причина такого различия это то, что задача кластеризации очень плохо поддается формализации. В то время как существуют объективные и достаточно точные методы оценки качества

классификации, оценка качества кластеризации, как правило, основывается на мнении эксперта и трудновыразима в численных показателях. Другими словами, одной из фундаментальных проблем кластеризации текстовых документов является оценка качества полученных результатов, так как не существует единого, общепризнанного и применимого во всех случаях метода оценки. [4]

1.2 Обзор существующих техник кластеризации текстов

В общем случае задача кластеризации основывается на метриках близости. Исходные документы представляются в виде вектора в пространстве определенных признаков. Подходы, для формирования вектора признаков могут существенно отличатся друг от друга и метрики могут высчитываться различными способами и являются отдельной задачей. В простейшем случае каждый признак соответствует наличию слова или словосочетания в исходном тексте. Величина компоненты может определятся также различными способами, например, компонента может быть истиной (или единицей) если рассматриваемое слово/словосочетание присутствует в данном тексте или нулем в противоположном случае; величина может рассчитываться по количеству вхождений рассматриваемого слова в документ (частота встречаемости) или рассчитываться какими-либо другими более сложными формулами, например, учитывать среднюю встречаемость конкретного слова по текущему набору текста относительно всего корпуса документов. Мера близости между текстами в данном случае будет рассчитываться как скалярное произведение между векторами. Большинство алгоритмов кластеризации в качестве исходных данных используется прямоугольную матрицу S, которая составлена из векторов документов и квадратной матрицы близости (формула 1.1).

$$V = S * S(t) \qquad (1.1)$$

Основным недостатком методов, использующих такую матрицу, является слишком большая размерность пространства признаков, некоторые из которых

являются избыточными и могут сказаться на точности результата (маскировать сходство между документами при его отсутствии).

Один из основных методов кластеризации является иерархическая кластеризация. Такие алгоритмы кластеризации строят на выходе дендрограмму (бинарное дерево), которое связывает все тексты. Существуют различные вариации таких алгоритмов, которые могут строить дерево как сверху вниз (рассматривающие исходный набор текстов как один единый кластер), так и снизу вверх (рассматривающие каждый текст из набора как один кластер). Таким образом, мы имеем все сечения бинарного дерева, показывающее итоговые кластеры. Ярким примером таких алгоритмов могут являться Single Link, Complete Link, Group Average. [5] Данные названия описывают, каким образом, будут определяться расстояния между кластерами:

- single Link – минимальное расстояние между парой объектов в соседних кластерах;
- complete Link – максимальное расстояние между парой;
- group Average – среднее расстояние.

Основным недостатком данного метода является то, что в общем случае мы получаем полную дендрограмму, то есть полное бинарное дерево по всем текстам. И количество кластеров в данном случае, так или иначе должно задаваться явно, чтобы наглядно увидеть срез дендрограммы.

Второй группой методов является неиерархическая кластеризация. Самыми распространенными алгоритмами в данной группе являются KMeans и EM (Expectation Maximization). В простейшем варианте алгоритму KMeans требуется задание начальных положений центроидов и числа кластеров, после чего запускается итеративный процесс, который стабилизирует центроиды. На каждом шаге документы приписываются к кластеру с ближайшим центроидом. После того как все тексты распределятся, будет высчитано новое положение центроидов. Если центроиды перестали перемещаться или достигнуто условие окончания процесса, то считается что кластеризация выполнена. Данный алгоритм практически всегда используется с вспомогательными алгоритмами, которые могут найти оптимальное

положение начальных центроидов и количество кластеров. Для вычисления положения начальных центроидов используется алгоритм Single/Average Link. В данном случае, размер случайной выборки рассчитывается по формуле 1.2:

$$V = \sqrt{k * n} \quad (1.2)$$

где k – число кластеров,

n – количество исходных текстов.

А для вычисления оптимального числа кластером можно использовать алгоритм Minimun Description Length (MDL). KMeans один из самых простых и широкоиспользуемых алгоритмов кластеризации, также является частным случаем общего метода EM. Метод Expectation Maximization работает с вероятностной моделью определения исходного документа к какому-то определенному кластеру. Вектора документов, в данном случае, рассматриваются как случайная величина. Если заранее известны параметры распределения, тогда можно вычислить условную вероятность принадлежности вектора к кластеру. [6]

Основным недостатком данного семейства алгоритмов является то, что для текстовых данных не всегда можно определить параметры распределения и метод не гарантирует достижение глобального минимума СКО.

Еще один метод, который используется для кластеризации текстов является метод анализа основных компонент (PCA). Он использует диагонализацию полной ковариационной матрицы термов. Данный метод обладает очень не высокой скоростью работы, поэтому применение его на сколь больших наборах текстов вряд ли возможно.

1.3 Обзор существующих техник кластеризации текстов с учетом семантической информации

Существуют различные попытки кластеризации текстов с учетом их семантической информации. Данная задача возникает тогда, когда мы имеет текстовую информацию, сформулированную разными словами и их порядком, но

несущую в себе схожий смысл, или схожими словами, но несущий в себе противоположный смысл. В таком случае, алгоритмы основанные на чистом лексическом сходстве могут не дать верный результат. Такие задачи решаются с помощь следующих способов [8]:

- составление тезаурусов;
- алгоритмический способ, устанавливающий и учитывающий ассоциативно-семантические связи между словами (LSA, pLSA, LDA).

Рассмотрим оба способа подробнее. Первый вариант – формирование тезаурусов достаточно трудоемкий и очень сильно зависит от специфики самой задачи. Создать универсальный тезаурус, который можно будет использовать для кластеризации любых текстов не представляется возможным на данный момент. Широкоиспользуемым тезаурусом является база понятий Википедия. Но в таком случае, во первых, мы полагаемся полностью на внешний источник данных, во-вторых, доверяем точность накопленных знаний, людям, которые создают данный портал. Также критичным недостатком является то, что близость понятий в данном случае будет определятся по близости терминов относительно друг друга в контексте тезауруса, а не в контексте исходного текста. Второй вариант – алгоритмический. Распространенным алгоритмом решения данной задачи является алгоритм латентно-семантического анализа (LSA). Данный метод применяется для рекомендательных систем, информационного поиска, кластеризации и еще ряда задач. Алгоритм находит скрытые смысловые взаимосвязи между объектами (объекты могут быть абсолютно любыми). Рассмотрим данный алгоритм на примере – у нас есть набор документов и мы хотим находить попарно близкие документы по смыслу. Вывод о близости мы можем делать на основе того, какие слова и как часто встречаются в этих документах. Аналогично классическим методам необходимо выполнить предобработку данных, выделить необходимые слова и сформировать таблицу признаков. Для подготовки данных, в самом простом случае можно использовать следующий подход – будем учитывать только частоту встречаемости слов. Предположим, что каждая тема характеризуется определенным набором слов и частотой. Если в тексте конкретный набор слов употребляется с определенными

частотами, то текст принадлежит к определенной теме. Порядок слов и морфологические формы для нас будут не важны. Таким образом, строится таблица слово-документ. В ячейках будет хранится либо 0 либо 1 в зависимости от отсутствия/наличия слова в тексте. В строках будут слова, а столбцы будут соответствовать документам из общего набора. Можно использовать различные метрики, не обязательно бинарную. Также распространено учитывать частоту слова в документе относительно общего набора документов (tf-idf). Далее для сравнения текстов вводится мера схожести двух столбцов таблицы (манхэттенское расстояние, евклидово, косинусное и тд). Далее полученная матрица раскладывается методом SVD (формула 1.3):

$$A = U * V * WT \quad (1.3)$$

Далее происходит выделение строк матрицы U и столбцов W, которые соответствуют наибольшим сингулярным числам (их может быть от 2-х до минимума из числа терминов и документов). Конкретное количество учитываемых собственных чисел определяется предполагаемым количеством семантических тем в задаче. А вообще чем больше сингулярное число, тем сильнее в коллекции проявлена тема.

Алгоритм LSA имеет ряд ограничений – семантическое значение документа определяется набором слов, которые как правило идут вместе, полностью игнорируется порядок слов, каждое слово имеет единственное значение. Основными же недостатками алгоритма LSA является то, что мы предполагаем о том, что карта слов не имеет вид нормального распределение (или имеет какое-либо распределение в различных вариациях и улучшениях алгоритма) и то, что эти алгоритмы имеют достаточно большое количество параметров, определение которых, может существенно повлиять на качество получаемого результата и определяется эмпирически.

2 Методика кластеризации текстов с использованием семантической информации

Задачу кластеризации с использованием семантической информации можно разделить на следующие этапы:

- получение семантически значимой информации, решение задачи извлечения данных;
- представление полученной информации в формализованном виде;
- ыычисление меры расстояния между признаками;
- непосредственно выполнение кластеризации.

Далее каждый из этапов будет рассмотрен отдельно и более подробно.

2.1 Извлечение данных

Исходные тексты на естественном языке имеют неформальную структуру, то есть мы строим предложения основываясь на правилах естественного языка, используя все многообразие словаря языка. Определенные фиксированные конструкции, как в языках программирования отсутствуют, поэтому задача извлечения данных имеет большую актуальность и зачастую служит первым этапом обработки текстов. Так как в рамках данной работы задача извлечения данных является вспомогательной рассмотрим ее в объеме, достаточном для дальнейшего исследования. Извлечение информации (IE) – это вариант информационного поиска, связанный с выявлением сущностей и взаимосвязей между ними, при котором из неструктурированного текста выделяется структурированная информация, готовая для дальнейшей обработки. [8] Учитывая уровни представления естественного языка, процесс анализа текстовой информации можно разделить на следующие этапы, показанные на рисунке 1.1

Первым этапом является графематический анализ. Он необходим для разделения неструктурированного текста на предложения и слова. Графематический анализ включает в себя разделение исходного текста на слова и разделители,

выделение устойчивых словосочетаний, выделение имен собственных, выделение структурных элементов, выделение предложений из исходного текста.

Морфологический анализ выполняет нормализацию слов, то есть приведение слов к их начальной неизменяемой форме, и выделяет набор параметров приписанных к данной словоформе.

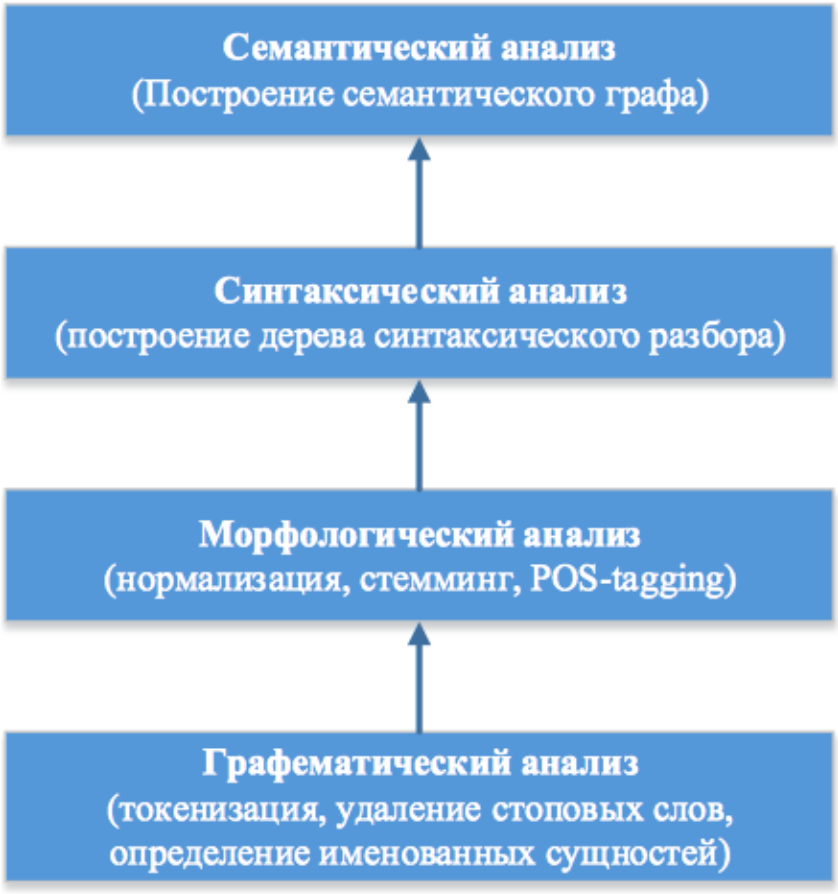


Рисунок 1.1 – Этапы анализа текстовой информации на естественном языке

Синтаксический анализ определяет роли слов и их взаимосвязи, в результате чего мы получаем дерево синтаксического разбора.

Семантический анализ ищет смысловые связи между выделенными понятиями, и основывается на результатах работы предыдущих анализаторов. На данном этапе появляется формальное представление смысла текста.

В результате работы анализаторов необходимо получить из полностью неструктурированной информации на естественном языке определенную, возможную для дальнейшей обработки электронно-вычислительными средствами, структурированную информацию. Оптимальным вариантом выглядит структура,

похожая на граф, в котором, буду отображаться сущности с их свойствами и взаимосвязи между этими сущностями. Формализованная структура представлена на рисунке 1.2.

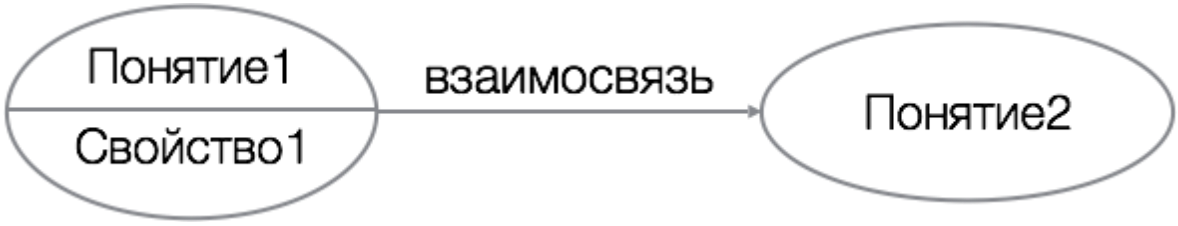


Рисунок 1.2 – Структура семантического графа

Для решения задачи извлечения фактов выделяют три основных подхода [9]:

- с использованием онтологий;
- машинное обучение (ML) ;
- с использованием формализованных правил.

Рассмотрим каждый подход более подробно.

Онтология включает в себя понятия и отношения между этими понятиями, описывает данные, которые необходимо извлечь из текста. Онтология это подробное описание области знаний. В контексте решения задачи извлечения данных в онтологиях описываются различные понятия, отношения между этими понятиями и их характеристики. Такой метод позволяет нам строить гипотезы по отношению к объектам в тексте и подтверждать их или отклонять.

Машинное обучение использует большие объемы входных данных и основывается на статистике. На основании вероятности появления той или иной лексической единицы в определенном контексте, система определяет ее в определенный результирующий набор. Основным недостатком данного подход является сложность обучения системы. Система очень сильно зависима от того, на каких наборах данных она обучена. Если входные тексты будут сильно отличатся от обучаемой выборки результаты работы алгоритмов будут некорректными. Для обучения алгоритмов используются структурированные тексты размеченные вручную и этот процесс сложно автоматизировать.

Подход, основанный на правилах, представляет собой написание шаблонов (регулярных выражений определенного типа) вручную на некотором

околоформальном языке. Эксперт составляет данные выражения, описывающие определенные факты в тексте, которые необходимо извлечь. И далее система работает по этим шаблонам. Типичными примерами можно выделить выделение имен собственных или дат.

Подводя промежуточные итоги, мы можем сделать вывод о том, что разработать алгоритм обработки естественного языка можно используя как машинное обучение, так и набор правил. Основным отличием является то, что для написания правил (шаблонов) необходимо привлекать эксперта, в то время как обучение системы с использованием машинного обучения практически не будет использовать человеческие ресурсы. Но подготовка размеченных корпусов может потребовать значительное время, которое превышает время, потраченное на написание шаблонов. [12]

2.2 Формализация данных

Для работы алгоритма кластеризации необходимо полученную структуру формализовать в таблицу. Чтобы решить данную задачу, необходимо составить вектор характеристик для каждого объекта. Это могут быть как числовые значения, например, возраст человека, частота встречаемости слов в тексте, рост или вес и так далее. Также могут быть категорийные характеристики (качественные). Остановимся на количественной характеристике как более оптимальной для решения нашей задачи, так как рассчитать по формуле вектор слова в тексте проще и логичнее, чем давать качественную характеристику. Таблица, как было сказано ранее, имеет следующий вид – в строках указаны номера текстов, по столбцам – уникальные слова, на пересечении частота встречаемости каждого слова в конкретном тексте. Частота встречаемости слова в тексте, на начальном этапе, будет считаться как элементарное количество вхождений слова в текст. Необходимо первоначально сравнить результат работы такой предобработки данных с обычной кластеризацией классическими методами. В качестве улучшения алгоритма расчёт будет производиться на основе структуры выделенного семантического графа. Семантической граф представляет

собой направленный граф, вершинами которого являются слова русского языка, отражающие основные понятия и их свойства, а ребра отражают взаимосвязи между основными понятиями. Структура графа была представлена на рисунке ранее. Частота встречаемости в данном случае будет рассчитываться следующим образом:

- полное совпадение рассматриваемого графа с графом текущего документа приравнивается к единице;
- если рассматриваемый граф полный (то есть имеется 2 понятия и взаимосвязь между ними), то в случае обратной взаимосвязи параметр будет равен 0;
- если рассматриваемый граф имеется в текущем документе, но либо основные понятия, либо их свойства (при наличии), либо взаимосвязь выражена словом синонимом, сохраняющем смысл, то каждый синоним отнимает от полного совпадения 0,2;
- в случае если рассматриваемый граф отсутствует в текущем документе, то параметр будет равен 0.

В результате будет получена формализованная таблица, готовая для обработки алгоритмами кластеризации (см. Таблица 2.1).

Таблица 2.1 – Формализованная таблица для алгоритма кластеризации

	Граф 1	Граф 2	Граф 3	Граф 4	…
Текст 1	0	1	0,8	0	
Текст 2	1	0,4	0,6	0	
Текст 3	0	0	1	0	
…					

2.3 Вычисление меры расстояния между признаками

Для работы алгоритма кластеризации необходим метод, по которому можно рассчитать близость объектов, то есть необходимо определенным образом ввести меру близости между признаками таблицы, полученной в предыдущем пункте. В качестве меры близости можно использовать различные формулы. Выбор данной формулы является отдельной задачей, на которой мы не будем останавливаться очень подробно. Рассмотрим часто используемые формулы, по которым рассчитывается близость между признаками. [10]

Евклидово расстояние - наиболее распространенная функция расстояния между признаками. Данное расстояние является геометрическим расстоянием в многомерном пространстве и рассчитывается по формуле (2.1).

$$p(x, x') = \sqrt{\sum_i^n (x_i - x_i')^2} \quad (2.1)$$

Квадрат евклидова расстояния – похоже на первую формулу, но придает больший вес более отдаленным друг от друга объектам. (формула 2.2)

$$p(x, x') = \sqrt{\sum_i^n (x_i - x_i')^2} \quad (2.2)$$

Манхэттенское расстояние – расстояние, являющееся средним разности по координатам. Работает похожим образом с евклидовым расстоянием и дает схожие результаты, но влияние отдельных больших выбросов уменьшается. (формула 2.3)

$$p(x, x_i) = \sum_i^n |x_i - x_i'| \quad (2.3)$$

Расстояние Чебышева – используется в тех случаях, когда нужно определить различие объектов. Факт различия устанавливается при различии по какой-либо одной координате. (формула 2.4)

$$p(x, x') = \max(|x_i - x_i'|) \quad (2.4)$$

Степенное расстояние – данное расстояние увеличивает или уменьшает вес для размерности, объекты которых сильно отличаются друг от друга. (формула 2.5)

$$p(x, x') = \sqrt[r]{\sum_i^n (x_i - x_i')^p} \quad (2.5)$$

где r и p – параметры, задаваемые пользователем.

Если эти параметры совпадают то формула преобразуется в расстояние Евклида. Параметр p – отвечает за постепенное взвешивание разностей по отдельным координатам, r – ответственен за прогрессивное взвешивание больших расстояний между объектами.

Косинусная мера – для двух векторов это косинус угла между ними. Помогает выявить пропорциональное сходство. (формула 2.6)

$$\cos(\vec{x}, \vec{y}) = \frac{\vec{x} * \vec{y}}{||\vec{x}||_2 * ||\vec{y}||_2} \quad (2.6)$$

Выбор метрики, как было ранее сказано, по сути отдельная задача. Решение о том, какую метрику выбрать, лежит на исследователе. Можно сделать предположение о том, что оптимальным для нашей задачи будет использование манхэттенского расстояния и косинусного. Манхэттенское расстояние может дать хорошие результаты в случае несхожести текстов по смысловому содержанию, а косинусное

расстояние, предположительно, может дать более точные результаты, если тексты близки по семантике. Данные предположения будут проверены в следующей главе данной работы.

2.4 Кластеризация

В результате данной работы, мы хотим сравнить результаты кластеризации с использованием существующих алгоритмов с результатами, полученными с учетом семантической информации исходных текстов. На начальном этапе будем использовать алгоритмы иерархической кластеризации и алгоритм kmeans. Данные алгоритмы были выбраны ввиду того, что они легко и быстро реализуются. На данном этапе может использоваться любой алгоритм классической кластеризации. Предполагается установить применимость того или иного алгоритма экспериментально.

3 Реализация методики кластеризации текстов с использованием семантической информации

3.1 Общее описание модулей системы

Разрабатываемая система должна иметь модульную архитектуру. Это позволит системе быть открытой, гибкой, даст значительные преимущества в ее поддержке. В качестве модулей можно выделить два основных компонента системы: предобработка данных (обработка исходных текстов, представленных естественно-языковыми средствами, формализация данных и представление их в виде, возможным для дальнейшей обработки компьютерными средствами, вычисление мер близости сформированных признаков) и выполнение кластеризации (кластеризация подготовленного формализованного набора данных одним из существующих алгоритмов) и один вспомогательных опциональный компонент: модуль предварительного форматирования (форматирование начальных текстов в необходимый для предобработки данных формат). Разработчик, может использовать любые программные языки, парадигмы программирования, интегрированные среды разработки, но с учетом особенностей конкретно выбранных инструментов, о которых будет рассказано далее. Если абсолютно все компоненты системы будут разрабатываться самостоятельно без использования внешних инструментов то условно ограничений на конкретную реализацию нету. Обобщенная архитектура системы представлена на рисунке 3.1.

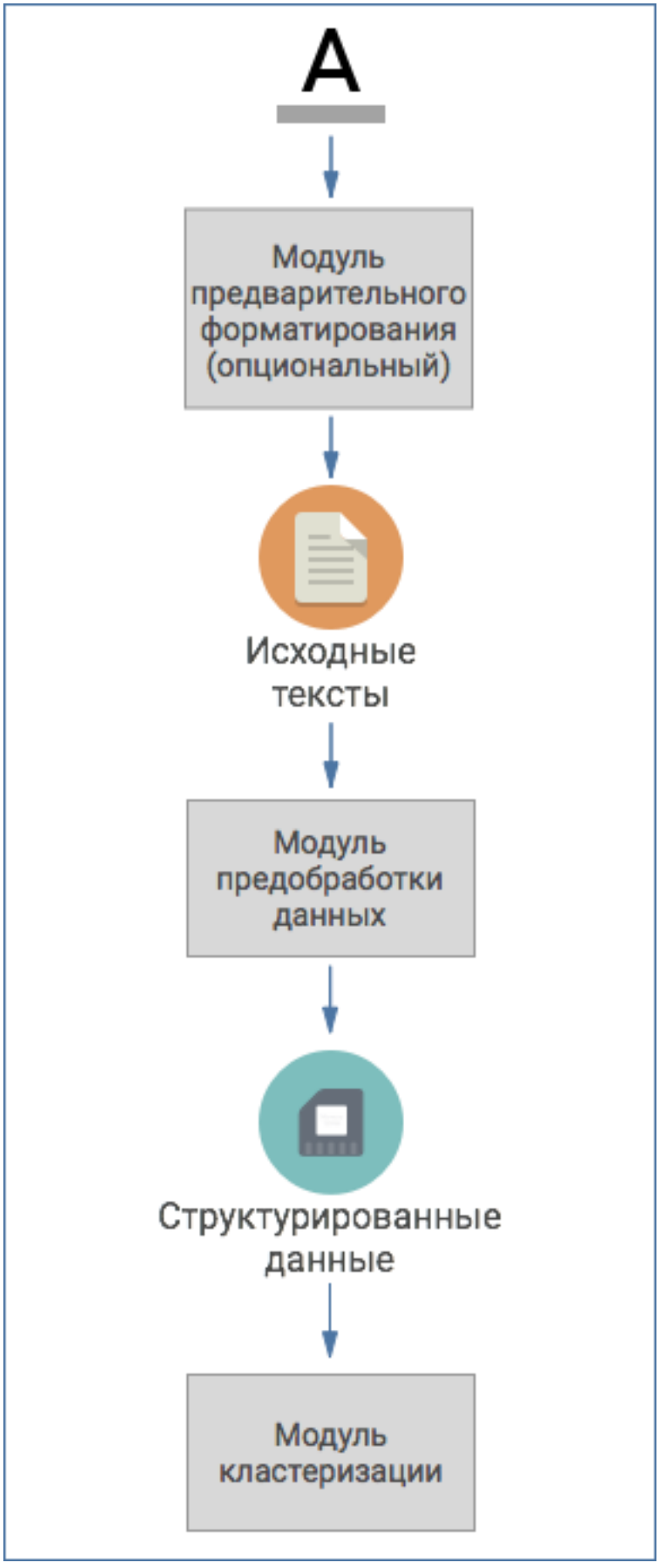


Рисунок 3.1 – Архитектура прототипа системы

Рассмотрим назначение каждого модуля на этой схеме более подробно:

- модуль предварительного форматирования данных (опциональный) – данный модуль форматирует документ в формат, необходимый для запуска логики предварительной обработки данных. Важно заметить, что на данном этапе не происходит никакой работы с содержимым документа. Изменяется только его расширение. Данный этап является необязательным в том случае, если исходные тексты уже отформатированы и готовы для запуска предобработчика;
- модуль предобработки данных – данный модуль, непосредственно, выполняет предобработку данных. Алгоритм работает с содержимым документа. На данном этапе, алгоритм преобразует информацию, представленную естественно-языковыми средствами, в формализованный набор данных, который можно будет обработать с помощью компьютера. На данном этапе происходят все описанные ранее шаги: выполняется графемологический анализ, морфологический анализ, синтаксический и семантический анализ. После работы данного алгоритма из текста будут удалены стоп слова, будет произведена токенизация, будет определена начальная форма слова, выполнен синтаксический разбор и выделена семантически значимая информация. Полученные данные будут иметь структуру описанного ранее графа и будут преобразованы в табличное представление для работы алгоритма кластеризации;
- модуль кластеризации данных – данный модуль, получив структурированные данные, выполнит кластеризацию документов любым доступным алгоритмом.

3.2 Модуль предварительного форматирования данных

Задача конвертации текстовых файлов достаточно проста. На сегодняшний день существует масса готовых сервисов, предлагающих решение данного вопроса. Можно использовать как готовый алгоритм с использованием API так и написать

собственный алгоритм. Не будем подробно останавливаться в рамках данной работы на этой задаче, так как научно-исследовательская сторона в ней отсутствует.

3.3 Модуль предобработки данных

3.3.1 Анализ существующих подходов к реализации алгоритма анализа по правилам

Логика данного модуля является основной в данной работе. От результатов работы данного модуля и решения задачи извлечения данных зависит качество кластеризации исходного набора документов поэтому остановимся на нем очень подробно.

Во второй главе, было решено и теоретически обосновано что для решения задачи извлечения данных мы будем использовать набор формализованных правил. Рассмотрим задачу извлечения данных и перечисленные ранее подходы на полностью практическом примере, чтобы лучше разобраться в поставленной задаче и определиться с используемыми инструментами.

Извлечение данных в данный момент широко используется поисковыми системами. Поисковые компании, таким образом, вытаскивают именованные сущности из текстов, строят аннотации к новостям, добавляют ссылки на схожие материалы и формируют карту. И все это происходит в автоматическом режиме без участия человека.

Рассмотрим задачу извлечения данных на нашем примере. Наша основная задача понять и учесть в кластеризации смысл рассматриваемых текстов. Человек, формирует свои мысли в предложения. Предложения на русском языке строятся определенным образом. Абсолютно четких правил как в английском языке нету и это заметно усложняет задачу, но в большинстве случаев основные понятия описываются существительными, их свойства выражены согласованными прилагательными, а взаимосвязи между словами выражаются глаголами. Также на смысл текста влияют

слова заключенные в кавычки, имена собственные, термины и сокращения. Нам необходимо их вытащить из текста в определенную, возможную для компьютерной обработки структуру.

Первое, и наверное самое простое решение, которое приходит в голову это использовать регулярные выражения. Но для решения задачи в общем виде, даже сейчас понятно, что такое регулярное выражение получиться очень сложным и практически не поддерживаемым. Также, очень существенным минусом будет то, что при малейшем отличии входных данных от написанного регулярного выражения, оно попросту не сработает. Регулярные выражения дадут отличные результаты если будет необходимо выделить из текста даты, денежные единицы или какие-либо численные данные. Аналогично их можно приспособить для написанных по всем правилам имен собственных. Но данная информация может и не содержать в себе семантику исходного текста. Нет необходимости экспериментально проверять этот метод, но не будем совсем от него отказываться. Мы вернемся к нему немного позднее.

Следующий рассматриваемый вариант это использование онтологии или проще говоря по онтологии или словарю. Метод достаточно прост и относительно эффективен. Онтология включает в себя основные понятия и отношения между ними. Любое предложение исходного текста можно разобрать по существующей онтологии и выявить сходства объектов или их различия. Но в данном случае, под каждую предметную область исходных текстов необходимо будет сначала построить онтологию, а потом уже по построенной онтологии выполнять разбор. На данный момент не существует абсолютно полной онтологии, которая покрывает все тематики потенциально возможных текстов. Даже если пытаться использовать универсальные базы знаний, например Википедию, то мы будем основываться на близости статей Википедии, что не всегда будет отражать реально существующие семантические связи исходного текста. Также возникает проблема многозначности слов. В этом случае одному термину Википедии может соответствовать много статей и без дополнительной обработки будет не понятно какую именно статью выбирать в качестве более похожей по смыслу.

Третьим упоминаемым подходом является машинное обучение. Машинное обучение это класс методов, которые решают поставленную задачу не напрямую, а «обучаются» в процессе применения решений множества схожих задач. Машинное обучение использует математические методы из разных областей таких как статистика, численные методы, методы оптимизации, теорию графов, теорию вероятностей. Рассмотрим задачу обучения в объеме, достаточном для пояснения дальнейших суждений. [11] Имеется множество объектов и множество ответов. Эти два множества зависят друг от друга но зависимость не известна, а известна только конечная совокупность пар «объект-ответ», которая и называется обучающей выборкой. На основе этих данных нужно найти зависимость и получить алгоритм, который сможет для любого нового объекта относительно точно предсказать ответ. В машинном обучении выделяют различные способы обучения – это и возможность обучения с учителем, и без учителя, обучение с подкреплением, обучение с частичным привлечением учителя и так далее. В классическом виде задача кластеризации может решаться с помощью машинного обучения. Но для выявления семантических связей машинное обучение использовать затруднительно ввиду того, что, аналогично с методом по онтологиям, мы не можем учесть все параметры исходных текстов, иными словами если мы обучили систему хорошо работать на художественных текстах, на технической литературе она скорее всего будет работать неверно и понять почему так происходит будет очень затруднительно или порой вообще невозможно. Придётся заново переобучать систему, выбирать другие критерии, оптимизировать различные статистические параметры – «подгонять» систему под набор текстов.

Для решения поставленной задачи, а именно извлечения именованных семантически значимых сущностей, мы будем использовать алгоритм, который может работать по формализованным правилам. Одной из самых простых реализаций такого алгоритма являются регулярные выражения, который не удовлетворяет полностью требованиям нашей задачи. Другим примером такого алгоритма является GLR-парсер. GLR-парсер это усовершенствованный алгоритм LR парсера. Рассмотрим как работает LR анализатор и его отличия от GLR-парсера.

Синтаксический анализатор состоит из написанных по определенным и регламентированным правилам формальных грамматик, которые в свое время состоят из нетерминальных и терминальных символов:

- нетерминальных символ – объект, которые полностью существует в тексте. Это неизменяемый объект. Через него можно задать какие-то определенные фиксированные значения, например «США», «Apple», или же какие-нибудь сокращения – «пр.», «ул.», «б-р.», «пер.»;
- терминальный символ – объект, который задает какую-то сущность языка. Например, название государства (не какое-то определенное, а в общем виде), название компании, ФИО человека и т.д. [13]

LR парсер работает следующим образом: на вход ему подается входная строка, он имеет стек разбора и грамматику. Рассмотрим работу LR анализатора на простом примере. [14]

Дана грамматика, разбирающая последовательность арифметических действий вида X*X*X…

Res -> G

G -> X

G -> G*G

Входная строка: X*X

Начальное состояние: X*X

Операция и сдвиг: X *X

Операция и свертка по правилу G -> X

Операция и сдвиг: G* X

Операция и сдвиг: G*X

Операция и свертка по правилу G -> X

Операция и сдвиг: G*G

Операция и свертка по правилу G -> G*G

Операция и сдвиг: G

Операция и свертка по правилу Res -> G

Операция и сдвиг: Res

Конечное состояние анализатора достигнуто и процесс завершен. В данном примере условно описан алгоритм работы LR анализатора. Представим что в качестве грамматики у нас не операция * и операторы, а слова естественного языка или группы слов (прилагательные, существительные) с определенными пометами и возможностью рассмотреть специальные символы, знаки препинания и т.д. Такой инструмент можно было бы использовать для решения нашей задачи. Этим инструментом является GLR-парсер. Этот анализатор является улучшенной версией LR анализатора. Дело в том, что LR парсер не умеет решать многозначности и применяется для обработки текстов формализованных языков, например исходных текстов программ написанных на языках программирования. В то время как GLR парсер является «параллельным парсером» - он обрабатывает естественный язык и обрабатывает все возможные трактовки входной последовательности, используя поиск в ширину.

3.3.2 Обзор существующих систем GLR парсинга

Как было сказано ранее, разработчик может написать свой GLR анализатор с нуля, но такие решения предлагаются различными поставщиками услуг. Рассмотрим самые распространённые из них.

Томита-парсер – это инструмент для извлечения структурированной информации из текста на естественном языке, предлагаемый и используемые компанией Яндекс. Для извлечения фактов инструмент использует контекстно-свободные грамматики и словари ключевых слов. Сам парсер представляет собой инструмент без каких-либо преднастроек. Для решения нашей задачи парсеру необходимо отдать на вход исходный текст, конфигурационный файл, корневой словарь, грамматику и тип факта, который мы хотим извлечь. Корневые словари для русского языка компания Яндекс предоставляет. Соответственно для успешного решения нашей задачи нам необходимо написать грамматику. Грамматика состоит из шаблонов, написанных на внутреннем языке/формализме Томита-парсера. Эти

шаблоны описывают цепочки слов, которые потенциально могут встретиться в тексте, а также описывают конечное представление извлеченных фактов. [13]

JAPE – в вычислительной лингвистике, это механизм шаблонных выражения платформы JAVA. Данный продукт распространяется с открытым исходных кодом и является компонентом платформы по обработке естественного языка. JAPE это преобразователь конечного состояния, который использует аннотации на основе регулярных выражений. Данный инструмент можно использовать для поиска выражений по образцу, семантического извлечения данных, выполнения операций над синтаксическими деревьями. JAPE грамматика состоит из правил, которые представляют собой набор шаблонов/действий возможных в описываемом случае. В итоге мы имеем с одной стороны описание аннотации, с другой стороны операции, которые возможно совершить с данными аннотациями. [15]

AGFL – парсер для обработки естественного языка. Данный инструмент умеет решать неоднозначности в естественном языке. Грамматики пишутся на своем, внутреннем языке, который обладает определенной лексикой. Распространяется по лицензии GNU GPL. [16]

LSPL – это язык, предназначенный для формального описания выражений русского языка для их дальнейшего извлечения из текста и последующей обработки. Язык изначально был создан для автоматической обработки ряда научно-технических текстов. Конструкции описываются в виде лексико-синтаксических шаблонов, которые определяют наличие слова с учетом его морфологических и грамматических характеристик. В данный момент система используется для извлечения информации из русскоязычных текстов финансовых обзоров биржевых компаний, может автоматически строить глоссарии для научно-технических документов и используется в основе вопросно-ответной системе по теории элементарных чисел. [17]

Как мы видим все из перечисленных инструментов является технологией GLR анализатора. Сформируем основные критерии выбора готового инструмента для решения нашей задачи:

- наличие возможности обработки с русском языком;

- наличие документации и примеров использования;
- области применения инструмента должна быть схожей с нашей (анализ текстов, выделение именованных сущностей, решение задачи NLP).

Наиболее подходящими инструментами среди рассмотренных систем являются Томита-парсер и LSPL так как оба инструмента могут работать с русским языком, в данный момент поддерживаются, имеют документацию и позволяют извлекать структурированную информацию из текстов на естественном языке. AGFL-парсер в данный момент не поддерживается, официальный сайт не работает, а JAPE не работает с русским языком. Несмотря на огромное сходство инструментов Томита-парсер и LSPL в конечном итоге для решения задачи был выбран Томита-парсер, так как он обладает лучшей документацией, в том числе примеры его использования и видеоуроки по настройке инструмента. Сравнительный анализ инструментов представлен в таблице 3.1.

Таблица 3.1 – Сравнительный анализ GLR парсеров

Название инструмента	Применение	Поддерживаемые языки	Аудитория	Доступность	Поддержка
Томита-парсер	Извлечение структурированных данных из неструктурированных текстов	Русский	Разработчики в области NLP, исследователи, студенты	Бесплатно (Mozilla Public License)	Поддержка компанией Яндекс, документация и видеопримеры и примеры с исходным кодом
JAPE	Выделение именованных сущностей, обработка естественного языка	Английский	Ученые и разработчики в области NLP, студенты, бизнес	Бесплатно (GNU)	Поддержка через официальный сайт, обширная документация, примеры
AGFL	Обработка естественного языка	Английский	Разработчики в области NLP, исследователи, студенты	Не доступен	Не поддерживается, официальный сайт не работает
LSPL	Формальное описание конструкций русского языка с целью их представления в системах извлечения знаний из текстов	Русский	Ученые и разработчики в области NLP, студенты, бизнес	Бесплатно (GNU)	Поддержка через официальный сайт, наличие документации, отсутствие примеров

3.3.3 Использование инструмента Томита-парсер в качестве GLR анализатора

Рассмотрим извлечение данных из текста с помощью Томита-парсера. Для решения этой задачи нам необходимо выполнить определенную последовательность шагов:

- составить словарь ключевых слов;
- выделить цепочки слов (предложения), в которых встречаются слова из словаря;
- сформулировать и описать на формальном языке правила (грамматики) извлечения фактов из текста;
- по сформулированным грамматикам выделить необходимую информацию и представить ее в формальном виде.

Первый этап – составление словаря ключевых слов. Существуют различные способы формирования этого словаря. Основными выделяют следующие:

- экспертный;
- по онтологии;
- обработка исходного текста с учетом частоты встречаемости понятия и терминов.

Задача составления словаря ключевых слов является не тривиальной и не является предметом исследования в данной работе. Не будем на ней подробно останавливаться и потому, что Яндекс предоставляет словарь ключевых слов для своего инструмента. Соответственно мы будем использовать готовое решение.

Далее нам необходимо выделить цепочки слов, которые содержат в себе слова из словаря. Объединим этот этап со следующим – с составлением грамматик для извлечения фактов из текста. Дело в том, что эти этапы очень близки и второй напрямую связан с первым. Сначала нам необходимо с помощью конструкций языка парсера описать цепочки слов с целью дальнейшей обработки. Такой анализ позволяет выделить семантически значимые слова и предложения для дальнейшей обработки уже с использованием грамматик.

Данный этап является ключевым при обработке текста. От того насколько точно описаны цепочки слов будет зависеть качество работы анализатора. На данном шаге выполняется морфологический и синтаксический анализ. Морфологический анализ позволит нам решить проблему согласования слов в цепочках. Парсер во время морфологического анализа нормализует слова с помощью определенных граммем. Причем в конечном выводе данные можно представить как в нормализованном виде, так и в исходном. В основе синтаксического анализа лежат формальные грамматики, описанные на языке анализатора. По ним строиться дерево разбора, которое отражает структуру входных цепочек.

Формальную грамматику можно описать как совокупность объектов. Она включает в себя алфавит терминальных символов, набор нетерминальных символов, набор правил аналогичных обычному LR анализатору с тем условием, что с левой стороны находится непустая последовательность терминальных и нетерминальных символов, содержащих хотя бы один нетерминальный символ, а с правой стороны может быть абсолютно любая последовательность терминальных и/или нетерминальных символов, и также необходим стартовый символ грамматики из набора нетерминалов.

Упомянутые ранее правила задаются шаблонами на языке парсера. Шаблон формально описывает конструкцию естественного языка, которая будет искаться в тексте. Шаблон строится как последовательность элементов, описывающих соответствующие фрагменты языковой конструкции с учетом порядка. Средства языка позволяют задавать вариативность конструкции, включая набор входящих в нее слов (лексем) и их морфологических характеристик (признаков). Рассмотрим написанное выше на обобщенном примере.

В левой части указывается название шаблона, а в правой варианты формализуемой фразы. Типовое описание конструкции выглядит следующим образом:

NounAdjective = Noun Adjective <Noun=Adjective>

NounAdjective – название шаблона, описывающего конструкцию естественного языка из существительного и прилагательного после

Noun – существительное

Adjective – прилагательное

<Noun=Adjective> - условие согласования, задаваемое с помощью помет

Данный шаблон описывает грамматически согласованную именную группу из существительного и прилагательного, например – «Кот черный», «Облако белое».

На основе изученной документации и иных материалов, а также базовых принципов построения предложений в русском языке было принято решение составить набор выражений, которые смогут извлечь из текста семантически значимую информацию. Построенные выражения позволят извлечь из текста существительные в различных формах, которые обозначают предмет (понятие), согласованные прилагательные, которые обозначают признаки (свойства предмета) и глаголы, которые обозначают управление между понятиями. Рассмотрим выражения, составленные для извлечения основных понятий. Остальные выражения будут приведены в приложении 1 к данной работе:

а. simpleParameter->Noun;

б. namedParameter->Word<h-reg1>+;

в. quotedParameter->Word<l-quoted>+Word<r-quoted>;

г. currentWord->SimpleParameter interp (Object.Parameter);

д. currentWord-> NamedParameter interp (Object.Parameter);

е. currentWord-> QuotedParameter interp (Object.Parameter).

Для выделения параметра мы будем искать в тексте существительные (пункт а.), имена собственные, написанные с большой буквы и, возможно, состоящие из нескольких слов (пункт б.) и имена собственные, заключенные в кавычки (пункт в.). Последние три правила описывают интерпретацию выделенных ранее параметров в поля структурированного объекта. (г. д. е.). Интерпретация параметров будет рассмотрена немного позднее в работе.

Как мы видим из описанных шаблонов есть поддержка метасимволов и помет:

- \+ - непустое повторение описанных символов;
- * - повторение символов;
- <l-qoute> - левая кавычка.

Инструмент поддерживает очень большое количество специальных символов и помет. Подробно с ними можно будет ознакомиться в официальной документации. [13]

После того, как из текста будут выделены фрагменты, описанные ранее их необходимо интерпретировать в формальную структуру, готовую для дальнейшей обработке. Именно этим и занимаются правила (4-6). Для дальнейшей обработки было сформирована такая структура, точно описывающая каждый элемент данных.

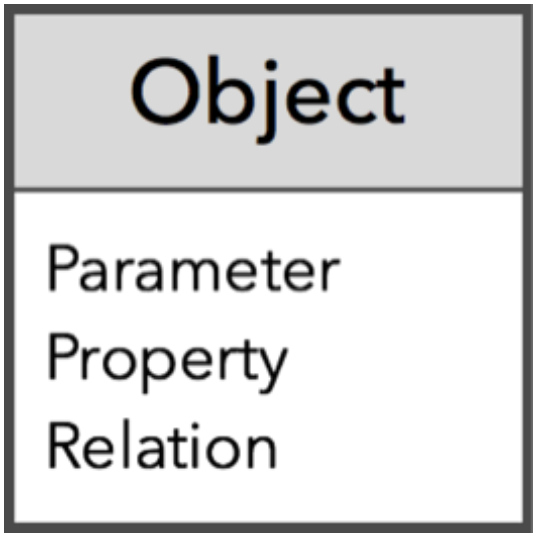


Рисунок 3.2 – Структура выделенного факта

В результате работы анализатора мы получим набор объектов представленных на рисунке 3.2. По своей сути объекты будут представлять логический граф. Сам объект характеризуется тремя полями – параметр, свойство и отношение.

Параметр отображает информацию о выделенном понятии. Может состоять как из одного слова, так и из нескольких.

Свойство отображает информацию о свойствах выделенного понятия. Данное поле опциональное и может отсутствовать.

Отношение отображает информацию об отношении данного понятия с другими понятиями. Поле также опционально.

Для большего понимания рассмотрим описанную структуру на простом примере.

У нас есть исходный текст:

«Для приготовления ризотто используется богатый крахмалом рис»

После обработки с использованием парсера из него будут выделены следующие данные:

Object1{

Parameter = «Приготовление ризотто»

Relation = «Используется»

}

Object2{

Parameter = «Рис»

Property = «Богатый крахмалом»

}

Как мы видим анализатор выделил из текста две сущности «приготовление ризотто» и «рис», которые связаны друг с другом через отношение «используется», и вторая сущность характеризуется свойством «богатые крахмалом». Таким образом мы имеем перед собой логическую структуру семантического графа, которую в дальнейшем может обрабатывать компьютер.

Техническую сторону решения проиллюстрируем с помощью рисунка 3.3. [9][13]

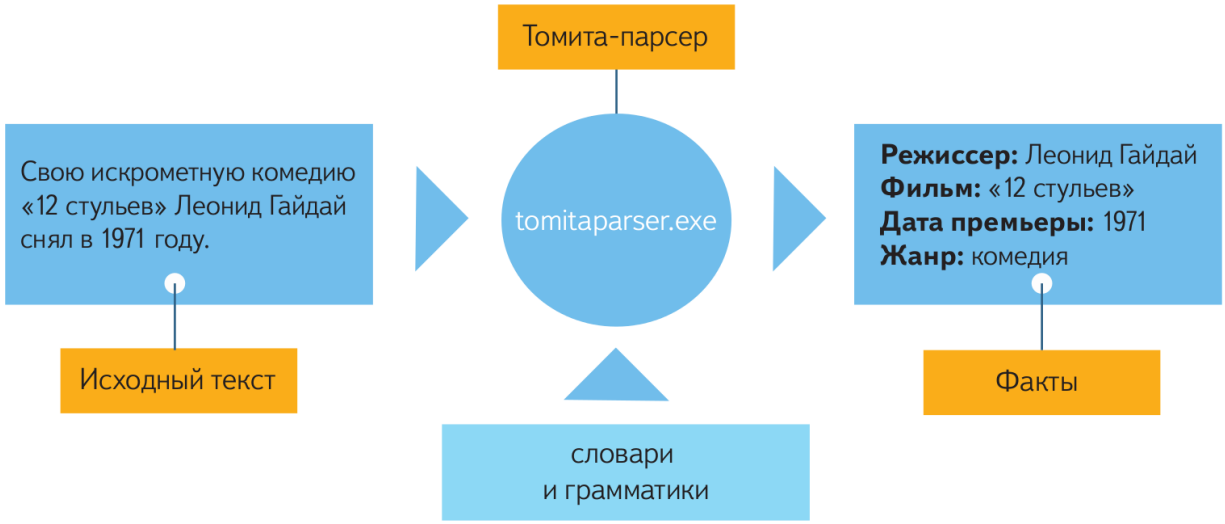


Рисунок 3.3 – Принцип работы инструмента Томита-парсер

На вход парсер получает исходный текст в формате txt. А также словарь ключевых слов и грамматики, сформированные пользователем. Дополнительно формируется файла факта, в который найденные данные будут интерпретироваться и конфигурационный файл. После того, как все необходимые файлы буду созданы из консоли запускается исполняемый файл, который проверит правильность исходных кодов и в случае успеха выполнит разбор текста.

Также рассмотрим вкратце исходные файлы простого проекта. Описание файлов представлено в таблице 3.2

Таблица 3.2 – Исходные файлы проекта для инструмента Томита-парсер

Имя файла	Комментарий
config.proto	Конфигурационный файл парсера. Сообщает парсеру, где искать все остальные файлы, как их интерпретировать и что делать. Нужен всегда
dic.gzt	Корневой словарь. Содержит перечень всех используемых в проекте словарей и грамматик. Нужен всегда
mygram.cxx	Грамматика. Нужен, если в проекте используются грамматики. Таких файлов может быть несколько.
facttypes.proto	Описание типов фактов. Нужен, если в проекте порождаются факты. Парсер запустится без него, но фактов не будет.
kwtypes.proto	Описания типов ключевых слов. Нужен, если в проекте создаются новые типы ключевых слов.

Конфигурационный файл является основным для запуска работы парсера. В нем содержится конфигурация инструмента, указываются пути, где искать остальные файлы, какие словари использовать и как интерпретировать найденные данные.

Файл грамматики формируется пользователем и содержит в себе множество правил на языке контекстно свободных грамматик. Данные правила запускаются всегда на одном предложении. Файл представляет собой набор шаблонов написанных на языке Томита-парсера. Шаблоны описывают в обобщенном виде цепочки слов, которые могут встретиться в тексте, а также определяют как интерпретировать данные в итоговом выводе.

Файл фактов представляет собой описание структуры объектов конечного вывода. Данный файл похож на класс или структуру в классическом видении объектно-ориентированного программирования.

3.4 Модуль кластеризации

После получения формализованного набора данных из текста на естественном языке нам необходимо эти данные свести в структуру, готовую к обработке алгоритмом кластеризации. Поскольку рассматриваемые алгоритмы кластеризации работают с пространством векторов, то форматом представления данных является таблица. По построенной таблице и рассчитанным параметрам этой таблицы будет высчитана мера близости между признаками и выполнится кластеризация.

Как было рассмотрено в теоретической части данной работы таблица в первой строке содержит в себе найденные уникальные графы, в первом столбце номера текстов. На пересечении рассчитывается частота встречаемости признака (графа) в текстах. Как было сказано, формирование данной таблицы может быть выделено в отдельную обширную задачу и исследование. Учитывая наши данные рассмотрим наиболее логичный вариант.

Частота встречаемости в нашем случае будет рассчитываться следующим образом:

- полное совпадение рассматриваемого графа с графом текущего документа приравнивается к единице;
- если рассматриваемый граф полный (то есть имеется 2 понятия и взаимосвязь между ними), то в случае обратной взаимосвязи параметр будет равен -1;
- если рассматриваемый граф имеется в текущем документе, но либо основные понятия, либо их свойства (при наличии), либо взаимосвязь выражена словом синонимом, сохраняющем смысл, то каждый синоним отнимает от полного совпадения 0,2;
- в случае если рассматриваемый граф отсутствует в текущем документе, то параметр будет равен 0;
- в случае если между графами есть минимальное сходство, то в ячейку ставиться значение равное 0,2.

Рассматривая данный алгоритм формирования таблицы перед нами встает задача поиска синонимов к понятиям. Рассмотрим эту задачу коротко, так как она не является ключевой в данной работе.

Выделяют два основных подхода поиска синонимов к какому-то заданному слову [18]:

- HITS алгоритм;
- нейронные сети.

HITS алгоритм опирается на существующую базу понятий и оценивает взаимосвязи между ними путем подсчета ссылок. Авторитетные страницы по определенной теме содержат большое количество пересекающихся и входных ссылок. Это обеспечивается HUB страницами, агрегирующими ссылки сразу на несколько авторитетных страниц. Таким образом, если по какому-либо определенному алгоритму подсчитать значения authority и hub получиться набор тематически связанных авторитетных страниц, соответствующих запросу. Авторитетные страницы это страницы, которые соответствуют запросу и имеющие больший ранг среди страниц данной тематики. Очень часто в качестве базы знаний используется открытая Википедия, а схожесть понятий оценивается путем подсчета гиперссылок. Показатели hub и authority считаются по следующим формулам (3.1 и 3.2 соответственно):

$$h_j = \sum_{i:(j,i)\in E} a_i \qquad (3.1)$$

$$a_j = \sum_{i:(i,j)\in E} h_i \qquad (3.2)$$

Далее решается задача поиска вершин в графе на основе найденных весов. Основным недостатками данного алгоритма являются низкое быстродействие и тот факт, что близость терминов будет определятся их близостью в рамках Википедия, что не всегда может отражать действительность.

Другим вариантом решения задачи являются нейронные сети. Остановимся на них подробнее. Принцип работы в общих словах достаточно простой – имеется большой текстовый корпус в качестве данных, которые подаются на вход, каждому слову сопоставляется вектор, который содержит в себе координаты слов на выходе. Изначально создается словарь из слов, которые подаются на вход, а затем

вычисляется векторное представление слов. Оно рассчитывается на контекстной близости (косинусной близости) - слова, встречающиеся в тексте рядом с одинаковыми словами (а следовательно, имеющие схожий смысл), в векторном представлении будут иметь близкие координаты векторов-слов. Полученные векторы-слов можно в дальнейшем использовать для машинного обучения или обработки текстов на естественном языке. Так как данный вариант представляет из себя нейронную сеть, то обучение должно производиться на большом исходном наборе данных. Существуют различные реализации данного метода – Word2Vec (Google) и RusVectores. [19, 20] Эти два инструмента являются яркими примерами реализации такой нейронной сети. Для обучения Word2Vec использовалась англоязычная база поисковых запросов пользователей Google, а для RusVectores Национальный Корпус Русского Языка и русская Википедия. Оба инструмента можно обучить на своих пользовательских наборах данных. Обучение системы является очень важным шагом и как было сказано ранее исходный корпус слов должен быть максимально большим. Этот метод имеет высокую производительность потому что достаточно один раз произвести обучение и в последующем получать результат за доли секунды. Также существенным плюсом является то, что используя перечисленные наборы слов для обучения мы не основываемся только на Википедии и не рассматриваем термины, подсчитывая количество ссылок. Оценка близости происходит по встречаемости слова в разных контекстах, что даст более точные результаты.

В данной работе мы будем использовать реализацию нейронной сети RusVectores так как она изначально обучена работать с русским языком. Алгоритмически оба инструмента имеют схожие реализации. Просто RusVectores имеет внешний программный интерфейс и изначально нацелен на работу с русским языком, а Word2Vec поставляется с англоязычной базой.

Таким образом, формируется таблица признаков (таблица 3.3) по описанному алгоритму для запуска алгоритма кластеризации. Если в тексте существует обратный подграф было решено выставлять в данном случае -1, чтобы усилить

различие между признаками, таким образом будет отражено не полное отсутствие графа в документе, а диаметральная противоположность.

Таблица 3.3 – Таблица признаков для алгоритма кластеризации

	Граф 1	Граф 2	Граф 3	Граф 4	…
Текст 1	-1	1	0,8	0	
Текст 2	1	0,4	0,6	0	
Текст 3	0	0	1	0	
…					

Имея данную таблицу мы можем запустить классический алгоритм кластеризации. В нашей реализации мы будем использовать иерархическую кластеризацию, алгоритм KMeans. Данные алгоритмы были рассмотрены в теоретической главе данной работы. В результате тестирования необходимо проверить качество кластеризации с помощью сформулированной методики и качество классической кластеризации и сделать выводы.

4 Тестирование разработанной системы

4.1 Исходные данные для экспериментов

Изначально каких-либо фиксированных требования к текстам не выдвигается. В качестве текстов для тестирования методики предлагается рассматривать информационно-значимые тексты – научно-популярные, художественные, новостные. К таким типам текстов можно отнести:

- научные статьи;
- художественные произведения;
- новостные статьи;
- статьи из социальных сетей;
- доклады;
- справочники;
- и др.

Каждый из перечисленных текстов имеет свои особенности, например в научно-популярных текстах каждая лексическая единица обозначает понятие или абстрактный предмет, текст очень сдержанный и локаничный, в нем нет «воды» и заспамленности. В то время как художественные произведения можно отметить обилием эпитетов, описаний, речевых диалогов главных героев. Статьи из социальных сетей могут содержать в себе огромное количество грамматических и орфографических ошибок, могут быть составленные профессиональными SMM-специалистами и иметь очень странную структуру, которая не будет подходить под предположение, выдвинутое во второй главе данной работы о том, как строятся тексты в русском языке. Новостные статьи зачастую могут содержать в себе эмоциональный окрас, могут рассказывать об одном и том же событии, но с разных точек зрения – одни издания могут положительно характеризовать событие, другие издания отрицательно используя схожий набор лексики и терминов. В данной главе работы необходимо сравнить работу алгоритмов на различных наборах текстов. Для тестирования будем использовать следующие наборы данных:

- научно-популярную подборку текстов на различные тематики;
- новостную подборку текстов на схожие тематики;
- краткие и лаконичные новостные заголовки.

Входными данными в задаче семантической кластеризации выступает текстовый файл в формате txt, в котором содержится набор неструктурированных текстов на русском языке. В ходе обработки и анализа набора текстов выполняются описанные ранее шаги и кластеризация исходных текстов.

4.2 Результаты экспериментов

Для объективности экспериментов и сравнения улучшенной кластеризации с классической начнем эксперименты с более простым алгоритмом формирования признаков. Таблица признаков будет составлена не с использованием графов, а с использованием конкретных слов и частоты их встречаемости. Эксперимент будет проведен на двух наборах данных – в первом случае возьмем семь текстов на различные темы, во втором случае возьмем новостные сообщения, очень схожие по смысловому содержанию, но разные по эмоциональному окрасу – проще говоря, одни новости будут описывать событие с положительной позиции, другие новости с отрицательной. Выполним кластеризацию двумя методами – иерархическую и Kmeans алгоритмом и сравним полученные результаты. В результате эксперимента у нас будет выполнена кластеризация четыре раза – два раза с использованием классических алгоритмов и два раза с использованием алгоритмов, с учетом семантической информации. Исходные тексты приведены в приложении 2, приложении 3.

Для расчёта метрики расстояния между документами будем использовать манхэттенское расстояние и косинусное (формулы описаны в теоретической главе данной работы). После подсчет расстояния можно запускать работу алгоритмов кластеризации. Исходными текстами в первом эксперименте являются тексты из приложения 2 – семь текстов из различных областей (1-«Ризотто», 2-«Паэльо», 3-«Google», 4-«Apple», 5-«Казанский собор», 6-«Исаакиевский собор», 7-«Microsoft»).

Количество слов в текстах сравнимо одинаковое. Стоит отметить тот факт, что при формировании таблицы признаков для алгоритмов кластеризации, таблица полученная с предобработкой данный при помощи Томита-парсера, содержит в себе наиболее значимую информацию исходного текста, в то время как чистое разложение текста в таблицу содержит абсолютно все найденные слова. Разница в размерах таблицы около 30%. Это говорит о том, что во время кластеризации будет использоваться только семантически значимая информация без случайных общих фактов, что также скажется на быстродействии в лучшую сторону. Экспериментально было установлено, что в качестве меры близости наиболее оптимально использовать Манхэттенское расстояние. Оно дало более стабильные результаты. В результате работы алгоритмов кластеризации мы ожидаем увидеть следующие логичные категории: архитектура, информационные технологии, кулинария. Результаты работы иерархической кластеризации представлены на рисунках 4.1 и 4.2.

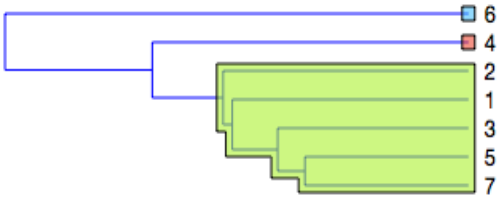


Рис 4.1 – Иерархическая кластеризация

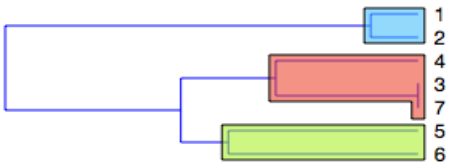
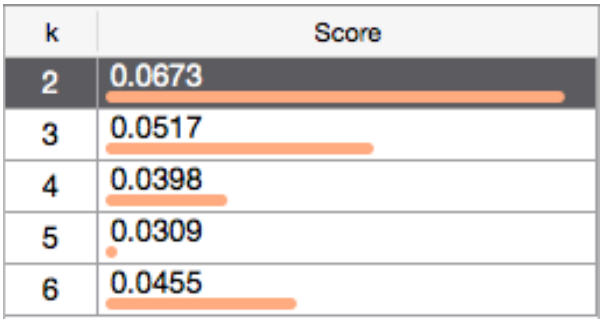


Рис 4.2 – Иерархическая кластеризация (семантическая)

Оценивая правильность работы алгоритма иерархической кластеризации мы видим, что точность работы алгоритма с учетом семантической информации выше, чем без нее. Кластеризация документов выполнена верно и при задании количества кластеров равное трем документы разбиты на верные корпуса.

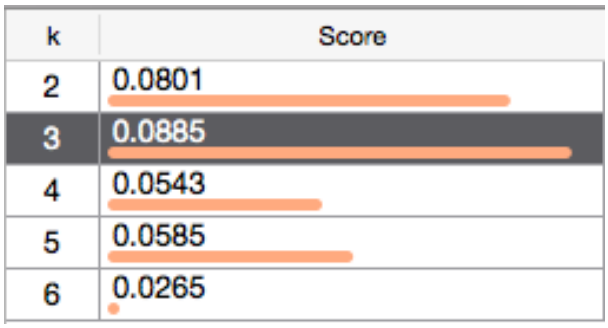
Аналогичные результаты были получены при использовании усовершенствованного алгоритма KMeans, который может рассчитывать оптимальное количество кластеров для исходных данных и выполнить кластеризацию (формулы также приведены в теоретической части данной работы). Результаты выделения кластеров представлены на следующих рисунках 4.3 и 4.4. Рисунки демонстрируют нам расчет оптимального количества кластеров по

исходным данным. В случае использования обычной кластеризации алгоритм считает оптимальным формирование двух кластеров, во втором случае, при учете семантической информации алгоритм считает оптимальным выделить три кластера, что является верным.



k	Score
2	0.0673
3	0.0517
4	0.0398
5	0.0309
6	0.0455

Рис 4.3 – Кластеры с использованием k-Means



k	Score
2	0.0801
3	0.0885
4	0.0543
5	0.0585
6	0.0265

Рис 4.4 – Кластеры с использованием k-Means (с учетом семантики)

Выполним такой же эксперимент для текстов, в которых смысл очень похож, но с точки зрения позиции описания абсолютно разный. Ярким примером таких текстов могут быть новостные сообщения. Разные издательства и государства могут иметь разные взгляды на одно и то же событие и выдавать его с выгодой для себя или же в текстах речь может идти про одни и те же события, например финансовая отчетность компании, но семантика может быть абсолютно разной – компания увеличила доходы в одной месяце, а в другом месяце ее доходы упали. Именно такие сообщения и будут исходными данными для эксперимента (приложение 3).

Экспериментально было установлено, что в качестве меры близости наиболее оптимально использовать косинусное расстояние. Оно дало более стабильные результаты, в случае если исходные тексты схожи по смысловому содержимому. В результате работы алгоритмов кластеризации мы ожидаем увидеть две логичные категории – первая, где новости позитивно описывают ситуацию и вторую – где,

новости характеризуют ситуацию с негативной точки зрения. Результаты работы иерархической кластеризации представлены на рисунках 4.5 и 4.6.

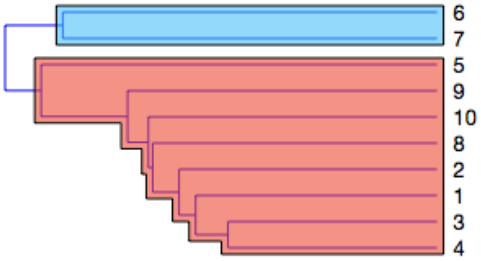


Рисунок 4.5 – Иерархическая кластеризация

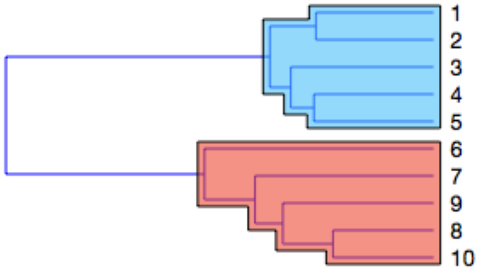


Рисунок 4.6 – Иерархическая кластеризация (семантическая)

Оценивая правильность работы алгоритма иерархической кластеризации мы видим, что точность работы алгоритма с учетом семантической информации выше, чем без нее. Кластеризация документов выполнена верно и при задании количества кластеров равное двум (положительный окрас, отрицательный окрас) документы разбиты на верные корпуса.

Аналогичные результаты были получены при использовании усовершенствованного алгоритма KMeans, который может рассчитывать оптимальное количество кластеров для исходных данных и выполнить. Результаты выделения кластеров представлены на следующих рисунках 4.7 и 4.8.

k	Score
2	0.0948
3	0.1059
4	0.0927
5	0.0920
6	0.0818

Рис 4.7 – Кластеры с использованием k-Means

k	Score
2	0.216
3	0.162
4	0.125
5	0.062
6	0.064

Рис 4.8 – Кластеры с использованием k-Means (с учетом семантики)

Рисунки демонстрируют нам расчет оптимального количества кластеров по исходным данным. В случае использования обычной кластеризации алгоритм считает оптимальным формирование трех кластеров, во втором случае, при учете

семантической информации алгоритм считает оптимальным выделить два кластера, что является верным.

Подводя промежуточный итог мы явно видим преимущество алгоритма с использованием семантической информации на различных данных. Проведем эксперименты с использованием графов, как это было описано ранее в работе.

Выполним алгоритм на различных текстах, как это было сделано ранее в работе и оценим полученные результаты. Исходные тексты представлены в приложении 2 и 3. В данном случае в таблице признаков в верхней строке будут подграфы документа. Тем самым таблица будет примерно на 45% меньше, чем исходная. Также важно отметить тот факт, что полученная таблица менее разреженная – то есть нулевых значений в ней существенно меньше, чем в подсчете слов по частоте встречаемости. Это происходит из-за того, что подграфы в текстах ищутся с учетом синонимов. Учитывается даже минимальное сходство в текстах, которое выражается десятыми.

Аналогично предыдущим экспериментам выполним кластеризацию с использованием графов и сравним результаты. (смотри рисунки 4.9 и 4.10)

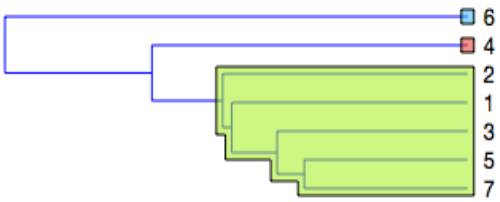


Рисунок 4.9 – Иерархическая кластеризация

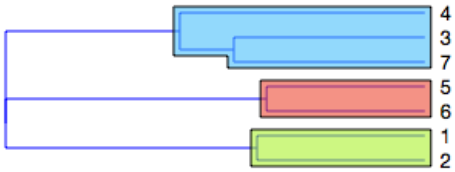


Рисунок 4.10 – Иерархическая кластеризация (семантическая с использованием графов)

Оценивая результаты работы алгоритма иерархической кластеризации мы видим, что выполнение алгоритма с использованием семантики исходных текстов дает верные результаты по сравнению с обычным классическим алгоритмом. Также можно отметить то, что семантический алгоритм с использованием графов формирует более равнозначные кластеры по сравнению с семантической кластеризацией без графов.

Рассмотрим результаты, полученные при использовании усовершенствованного алгоритма KMeans, который может рассчитывать оптимальное количество кластеров для исходных данных и выполнить кластеризацию (формулы также приведены в теоретической части данной работы).

k	Score
2	0.0673
3	0.0517
4	0.0398
5	0.0309
6	0.0455

Рисунок 4.11 – Кластеры с использованием k-Means

k	Score
2	0.227
3	0.299
4	0.221
5	0.108
6	0.090

Рисунок 4.12 – Кластеры с использованием k-Means с учетом семантики на графах)

На рисунках 4.11 и 4.12 явно видно, что семантическая кластеризация определяет оптимальным разбиение на три кластера, в то время как классический алгоритм предлагает разбиение на два кластера. Визуализированные результаты кластеризации алгоритма k-Means представлены в приложении 4 (1 и 2). Во втором случае кластеры выделены четко, расстояние между элементами в кластерах минимально.

Проведем такой же эксперимент на текстах, со схожим смыслом и сравним результаты обоих алгоритмов. Из исходных текстов было удалено два текста, которые выбивались из общего смысла и оставлены наиболее схожие по смыслу текстов. (исходные тексты представлены в приложении 3. Первые два текста были удалены).

Выполним иерархическую кластеризацию восьми текстов схожих по смыслу и оценим полученные результаты (смотри рисунки 4.13 и 4.14).

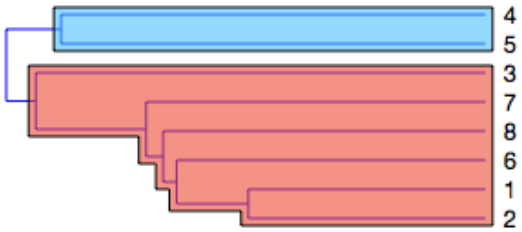


Рисунок 4.13 – Иерархическая кластеризация

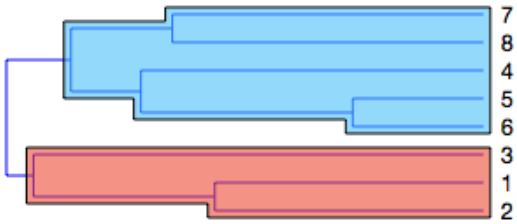


Рисунок 4.14 – Иерархическая кластеризация (семантическая с использованием графов)

Первые три текста положительно характеризуют экономическую обстановку, а остальные пять текстов – негативно. Семантическая кластеризация выдает подобные результаты. Классический алгоритм формирует неверное количество кластеров и разбиение.

Рассмотрим результаты, полученные при использовании усовершенствованного алгоритма KMeans, который может рассчитывать оптимальное количество кластеров для исходных данных и выполнить кластеризацию (формулы также приведены в теоретической части данной работы).

k	Score
2	0.1052
3	0.1074
4	0.0951
5	0.0833
6	0.0236

Рисунок 4.15 – Кластеры с использованием k-Means

k	Score
2	2.241
3	2.882
4	3.268
5	3.544
6	3.713

Рисунок 4.16 – Кластеры с использованием k-Means (с учетом семантики на графах)

На рисунках 4.15 и 4.16 явно видно, что семантическая кластеризация определяет оптимальным разбиение на два кластера, в то время как классический алгоритм предлагает разбиение на три кластера. Визуализированные результаты кластеризации алгоритма k-Means представлены в приложении 4 (3 и 4). Во втором случае кластеры выделены четко, расстояние между элементами в кластерах минимально.

Оценить качество кластеризации русскоязычных текстов достаточно сложно. Для иностранных текстов есть эталонная база, с которой можно сравнить полученные результаты. Вообще оценка качества кластеризации задача очень не простая. В данной работе результаты были оценены экспертным методом, как наиболее простым и точным, учитывая поставленную задачу. Стоит отметить следующий факт – в виду несовершенства словарей, который используются для разбора исходного текста на факты инструментом Томита-парсер, в некоторых случаях допускаются ошибки. Например прилагательное может быть интерпретировано, как существительное. Во время выполнения работы такие ошибки были отмечены примерно в 10% случаев (из 100 графов в 10 могут присутствовать неточности).

Для оценки качества кластеризации также можно использовать один из основных показателей индекс оценки силуэта. Данный индекс основан на вычислении величины силуэта для каждого образа, который определяет насколько данный объект кластера схож с объектами этого кластера насколько он отличается от образов других кластеров.

Индивидуальный индекс силуэта для образа $x^{(k,i)}$ рассчитывается по формуле 4.1.

$$r_{i,k} = \frac{(b_{i,k} - a_{i,k})}{\max(b_{i,k}, a_{i,k})} \quad (4.1)$$

где $a_{i,k}$ - среднее расстояние от $x^{(k,i)}$ до образов своего класса,

$b_{i,k}$ - минимальное среднее расстояние от $x^{(k,i)}$ до образов других классов

Просуммировав полученные индивидуальные индексы можно получить суммарный силуэт. Чем выше данное значение, тем качественнее выполнена кластеризация. Результаты индексов силуэта сведены в таблицу 4.1.

Таблица 4.1 – Суммарные индексы силуэта (Silhouette index)

Без использования графов	Обычный алгоритм	Семантический алгоритм
Разные тексты	0,59	1,96
Схожие	0,87	2,54
С использованием графов		
Разные тексты	0,59	2,24
Схожие	1,30	2,89

Оценивая результаты, можно отметить, что алгоритм с использованием семантической информации показал себя лучше во всех случаях. Алгоритм с использованием графов выполнил кластеризацию качественнее, чем без графов.

ЗАКЛЮЧЕНИЕ

В данной работе была описана методика кластеризации текстов на русском языке с использованием семантической информации исходных текстов. Для решения задач, поставленных в работе, были использованы системный подход, междисциплинарный подход, а также методы логического анализа и синтеза.

В ходе выпускной квалификационной работы были выполнены следующие задачи:

- изучена предметная область, проведен обзор существующих методов кластеризации, выявлены их преимущества и недостатки;
- исследованы методы извлечения данных из текстов на естественном языке;
- предложена методика выполнения кластеризации текстов на русском языке с использованием семантической информации;
- разработан прототип программной системы для выполнения семантической кластеризации;
- реализована и экспериментально проверена методика семантической кластеризации текстов;
- технически реализованы модули предварительного форматирования и предобработки исходных текстов.

Основные модули прототипа были реализованы в виде консольных приложений, для выполнения кластеризации был использован программный продукт Orange, который содержит реализации основных алгоритмов кластеризации и их визуализацию. Реализованная методика была опробована на научно-популярных текстах и новостных сообщениях различного и схожего смыслового содержания. Испытание методики позволяет заключить, что предложенный подход позволяет выполнять кластеризацию текстов на русском языке различных стилей в полуавтоматическом режиме. Разработанную методику можно использовать для разбиения существующих наборов текстов на кластеры, выполнять информационный поиск, алгоритм можно использовать в тех случаях, когда стандартных методов

кластеризации недостаточно и нужно обратить внимание на смысловое содержимое текстов.

В качестве дальнейшего развития данной работы, могут быть выбраны следующие направления:

- доработка модуля предобработки исходных текстов путем формирования дополнительных правил извлечения фактов, которые смогут покрыть больше потенциальных полей фактов;
- обход существующих ограничений использованного инструмента Томита-парсер (или использование собственного GLR анализатора);
- дальнейшее более глубокое экспериментальное исследование предложенной методики по сравнению с другими реализациями алгоритмов кластеризации текстов.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. А. Беленький «Текстомайнинг. Извлечение информации из неструктурированных текстов» [Электронный ресурс] – http://compress.ru/article.aspx?id=19605. Дата обращения: 20.05.2017
2. Nicholas O. Andrews, Edward A. Fox «Recent Developments in Document Clustering», English, 2007
3. П. С. Шеменков «Разработка и исследование модели нейросетевого метода анализа текстовых документов» [Электронный ресурс] – http://www.dissercat.com/content/razrabotka-i-issledovanie-modeli-neirosetevogo-metoda-analiza-tekstovykh-dokumentov. Дата обращения 20.05.2017
4. М. В. Киселев, В.С. Пивоваров, М.М. Шмулевич «Метод кластеризации текстов, учитывающий совместную встречаемость ключевых терминов, и его применение к анализу тематической структуры новостного потока, а также ее динамики» » [Электронный ресурс] – http://elar.urfu.ru/bitstream/10995/1421/1/IMAT_2005_22.pdf. Дата обращения 20.05.2017
5. Cambridge University Press, «Single Link, Complete-Link and Average-Link Clustering» - Introduction to Information Retrieval, English, 2011
6. К. М. Кириченко, М. Б. Герасимов «Обзор методов кластеризации текстовой информации» [Электронный ресурс] – http://www.dialog-21.ru/en/digest/2001/articles/kirichenko/. Дата обращения 20.05.2017
7. Habrahabr, «Кластеризация текстовых документов по семантическим признакам» [Электронный ресурс] – https://habrahabr.ru/post/324540/. Дата обращения 20.05.2017
8. Брюхов Д.О., Скворцов Н.А. Извлечение информации из больших коллекций русскоязычных текстовых документов в среде Hadoop // Труды 16-й Всероссийской научной конференции «Электронные библиотеки: перспективные методы и технологии, электронные коллекции» RCDL'2016. Дубна: ОИЯИ, 2014

9. Блог компании Яндекс. Извлечение объектов и фактов из текстов в Яндексе. Лекция для Малого ШАДа: [Электронный документ]. – (https://habrahabr.ru/company/yandex/blog/205198/). Дата обращения 20.05.2017
10. А.Часовских «Обзор алгоритмов кластеризации данных». [Электронный документ]. – https://habrahabr.ru/post/101338/. Дата обращения 20.05.2017
11. Hastie, T., Tibshirani R., Friedman J. The Elements of Statistical Learning: Data Mining, Inference, and Prediction. – 2nd ed. – Springer-Verlag, English, 2009
12. Зубарев В.С. Исследование и разработка алгоритма семантической информационно-поисковой системы: [Электронный документ]. – (http://storage.vas3k.ru/files/Main.pdf). Дата обращения 20.05.2017
13. Томита-парсер. Руководство разработчика. Версия 1.0: [Электронный документ]. – https://tech.yandex.ru/tomita/doc/dg/concept/about-docpage. Дата обращения 20.05.2017
14. Lections on Computer Science: LR-Parsing: [Электронный документ]. – http://math.msu.su/~vvb/BMSTU/lectLR.html. Дата обращения 20.05.2017
15. JAPE Developing Language Processing Components with GATE Version 8 [Электронный документ]. – https://gate.ac.uk/sale/tao/index.html#x1-1880008. Дата обращения 20.05.2017
16. AGFL Description [Электронный документ]. – https://nlnet.nl/project/agfl/description.html. Дата обращения 20.05.2017
17. Описание языка LSPL (1.0.1): [Электронный документ]. – (www.lspl.ru/articles/LSPL_Refguide_13.pdf). Дата обращения 20.05.2017
18. А.А. Крижановский «Автоматизированное построение списков семантически близких слов на основе рейтинга текстов в корпусе с гиперссылками и категориями» [Электронный документ]. – https://arxiv.org/pdf/cs/0606128.pdf. Дата обращения 20.05.2017
19. Word2Vec [Электронный документ]. – https://code.google.com/archive/p/word2vec. Дата обращения 20.05.2017
20. RusVectōrēs: дистрибутивные семантические модели для русского языка [Электронный документ]. – http://rusvectores.org/ru. Дата обращения 20.05.2017

21. Блог Вастрик.ру «Извлечение фактов из текста. Томита-парсер Яндекса» [Электронный документ]. – http://vas3k.ru/blog/354/. Дата обращения 20.05.2017
22. Блог Вастрик.ру «Извлечение фактов из текста. Дубль два» [Электронный документ]. – http://vas3k.ru/blog/358/. Дата обращения 20.05.2017
23. Блог компании АОТ «Автоматическая обработка текста» [Электронный документ]. – http://aot.ru/docs/seman.html. Дата обращения 20.05.2017
24. Orange Blog «Data MiningFruitful and Fun» [Электронный документ]. – https://blog.biolab.si. Дата обращения 20.05.2017
25. Biber, D., Conrad, S., Reppen, R., Aitchison, J., ‘Corpus Linguistics: Investigating Language Structure and Use’, Cambridge Univ. Press, English 1998
26. Fellbaum, C. (editor), ‘*WordNet: An Electronic Lexical Database*’, MIT Press, English, 2005
27. Саламаха О. «Алгоритм LSA для поиска похожих документов» [Электронный документ]. – https://netpeak.net/ru/blog/algoritm-lsa-dlya-poiska-pohozhih-dokumentov. Дата обращения 20.05.2017

ПРИЛОЖЕНИЕ 1

Исходный текст настройки Томита-парсера

Файл config.proto

```
encoding "utf8";
TTextMinerConfig {
    Dictionary = "dic.gzt";
    Input = {File = "5.txt"}
    Output = {File = "out5.txt"
                Format = text}
    Articles = [
             {Name = "сущ"}
             ]
    Facts = [{Name = "Object"}]
    PrettyOutput = "pretty.html"
}
```

Файл dic.gzt

```
encoding "utf8";
import "base.proto";
import "articles_base.proto";
import "fact_types.proto";

TAuxDicArticle "сущ"
{
    key = {"tomita:noun.cxx" type=CUSTOM}
}
```

Файл fact_types.proto

```
import "base.proto";
import "facttypes_base.proto";

message Object: NFactType.TFact {
    required string Parameter = 1;
    optional string Relation = 2;
    optional string Property = 3;
}
```

Файл noun.cxx

```
#encoding "utf8"
MyWordN -> Noun;
MyWordNoun -> Word<h-reg1>+;
```

```
     MyWordNoun2 -> Word<l-quoted>+ Word<r-quoted>;
     MyWordV -> (Noun) Verb;
     MyWordA -> Adj;

     //Verb
     MyWord -> MyWordN interp (Object.Parameter) MyWordV
interp (Object.Relation);
     MyWord -> MyWordNoun interp (Object.Parameter) MyWordV
interp (Object.Relation);
     MyWord -> MyWordNoun2 interp (Object.Parameter) MyWordV
interp (Object.Relation);

     //Noun
     MyWord -> MyWordN interp (Object.Parameter);
     MyWord -> MyWordNoun interp (Object.Parameter);
     MyWord -> MyWordNoun2 interp (Object.Parameter);

     //Adj
     MyWord -> MyWordN<gnc-agr[1]> interp (Object.Parameter)
MyWordA<gnc-agr[1]> interp (Object.Property);
     MyWord -> MyWordA<gnc-agr[2]> interp (Object.Property)
MyWordN<gnc-agr[2]> interp (Object.Parameter);
```

ПРИЛОЖЕНИЕ 2

Исходные тексты на различные темы

Текст 1

Ризотто – распространённое блюдо из риса в Северной Италии. Первое письменное упоминание о нём встречается только в XIX веке.

Для ризотто используется круглый богатый крахмалом рис сортов Арборио, Бальдо, Падано, Рома, Виалоне Нано, Марателли или Карнароли (последние три сорта считаются лучшими для ризотто и они наиболее дорогие).

Текст 2

Паэлья – национальное испанское (валенсийское) блюдо из риса, подкрашенного шафраном, с добавлением оливкового масла. Кроме этого в паэлью могут добавляться морепродукты, овощи, курица. Название происходит от латинского слова patella — «сковорода». Популярность этого блюда в настоящее время обусловлена множеством вариаций в ингредиентах, адаптированных к различным регионам испанской кухни.

Текст 3

Apple – американская корпорация, производитель персональных и планшетных компьютеров, аудиоплееров, телефонов, программного обеспечения. Один из пионеров в области персональных компьютеров[9] и современных многозадачных операционных систем с графическим интерфейсом. Штаб-квартира – в Купертино, штат Калифорния.

Текст 4

Google – американская транснациональная публичная корпорация, компания в составе холдинга Alphabet, инвестирующая в интернет-поиск, облачные вычисления и рекламные технологии. Google поддерживает и разрабатывает ряд интернет-сервисов и продуктов (Список сервисов, инструментов Google) и получает прибыль в первую очередь от рекламы через свою программу AdWord

Текст 5

Казанский кафедральный собор – один из крупнейших храмов Санкт-Петербурга. Построен на Невском проспекте в 1801—1811 годах архитектором

Андреем Воронихиным для хранения чтимого списка чудотворной иконы Божией Матери Казанской. После Отечественной войны 1812 года приобрёл значение памятника русской воинской славы. В 1813 году здесь был похоронен полководец Михаил Кутузов и помещены ключи от взятых городов и другие военные трофеи.

Текст 6

Исаакиевский собор – крупнейший православный храм Санкт-Петербурга. Расположен на Исаакиевской площади. Имеет статус музея (музейный комплекс «Государственный музей-памятник „Исаакиевский собор“»). Построен в 1818—1858 по проекту архитектора Огюста Монферрана; строительство курировал император Николай I, председателем Комиссии о построении собора был Карл Опперман. Творение Монферрана – четвёртый по счёту храм в честь Исаакия Далматского, построенный на этом месте в Санкт-Петербурге.

Текст 7

Microsoft Corporation – одна из крупнейших транснациональных компаний по производству проприетарного программного обеспечения для различного рода вычислительной техники — персональных компьютеров, игровых приставок, КПК, мобильных телефонов и прочего, разработчик наиболее широко распространённой на данный момент в мире программной платформы — семейства операционных систем Windows.

ПРИЛОЖЕНИЕ 3

Исходные тексты на схожую тематику

Текст 1

Россия готовит серию мощнейший атомных ледоколов. По заявлению «Росатома», «Арктика», "Урал" и "Сибирь" - самые большие и мощные атомные ледоколы в мире поступят в эксплуатацию в эксплуатацию в 2019 и 2020 годах. Легендарная «Арктика» - атомный ледокол, который первым достиг Северного полюса в надводном плавании и прославивший тем самым страну, его создавшую – СССР. Россия, будучи наследницей Советского Союза не только серьёзно отнеслась к достижениям техники, но и приумножила их, создав крупнейший в мире атомный ледокол и дав ему звёздное название - «Арктика». Построить такой смогли только в России на Балтийском заводе.

Текст 2

Мужская сборная России – чемпион мира в эстафете по биатлону. Сборная России по биатлону в составе Волков-Цветков-Бабиков-Шипулин стала чемпионом в эстафете на проходящем в Хохфильцене, Австрия, чемпионат мира. Сборная России прекрасно провела гонку с самого начала, постоянно находясь в лидирующей группе, и оторвалась от преследователей задолго до финиша.

Текст 3

В России невиданными темпами растет сельскохозяйственное машиностроение. Производство кормоуборочных комбайнов и энергонасыщенных тракторов в России увеличилось в 2016 году на 60%. Об этом сегодня министр промышленности и торговли Денис Мантуров доложил президенту РФ Владимиру Путину. «По зерноуборочным комбайнам – более 30%, по прицепной технике – 37 %. У нас последние три года такая положительная тенденция. За последние три года рост производства – в 2,5 раза.

Текст 4

Доходы Крыма в составе России выросли вдвое. Доходы полуострова выросли вдвое по сравнению с украинским периодом. За прошлый год Крым заработал 40,6 млрд рублей и это в два раза больше, чем в последние годы украинского периода.

Даже в самые лучшие времена в составе Украины бюджет Крыма составлял чуть больше 22 миллиардов рублей. И это - при относительной геополитической стабильности. Всего за три года, несмотря на все блокады и санкции, несмотря на сложный переходный период, Крым вышел на доходы бюджета, которые почти в два раза превышают украинские даже без учета федеральной помощи.

Текст 5

Российские трубные предприятия полностью обеспечили Газпром. В 2016 году отечественные трубные предприятия обеспечили полное импортозамещение труб большого диаметра для объектов корпорации Газпром. Основная часть продукции произведена для проектов «Сила Сибири» и «Ухта — Торжок — 2», всего более 800 тысяч тонн. Как сообщает Ассоциация производителей труб, инновационные разработки и новая продукция для объектов добычи (бурильные, насосно-компрессорные трубы и др.) допущена к использованию на объектах Газпрома, которые в 2016 году смогли отказаться от поставок немецких и японских поставок этих сортов продукции. Более того, по сообщению Ассоциации, российские трубные заводы теперь работают не только на импортозамещение: предприятия отрасли начали выпуск продукции, конкурентоспособной на зарубежных рынках.

Текст 6

ЦБ прогнозирует отрицательные темпы роста экономики России. Темпы роста экономики останутся на уровне -1%, считают в Центробанке.

Темпы роста российской экономики в течение ближайшего года останутся отрицательными - на уровне минус 1%, заявила глава Центробанка России Эльвира Набиуллина.

Текст 7

Экономика России летит вниз, но этот график заставляет Путина ликовать. Экономика России продолжает падать как камень из-за низких цен на нефть и экономических санкций. Ситуация не улучшится и в этом году, говорится в докладе известной московской Высшей школы экономики. Большинство секторов экономики испытывают проблемы – это, в частности, строительство, торговля и промышленность, но есть одно светлое пятно — сельское хозяйство.

Текст 8

12 апреля американское издание New York Times опубликовало свой взгляд на причины и процесс падения экономики России. Низкие цены на нефть и международные санкции подорвали экономику России. В стране бюджетный дефицит с 2012 года, и его резервный фонд закончится к 2017 году. Одной из причин нынешней экономической ситуации в России является резкое снижение мировых цен на нефть с июня 2014 года. В 2015 году нефть и газ приходилось 43% от доходов. Рубль упал почти на 50% по отношению к доллару с августа 2014 г. Это вызвало снижение уровня жизни по всей России, так как ослабление рубля делает импорт более дорогим. Россия противопоставила санкциям Запада запреты на импорт различных пищевых продуктов, что приводит к дальнейшему росту цен.

Текст 9

Снижение ВВП России восстановился в июле 2016 года, а если сравнить с аналогичным периодом прошлого года, то падение составило 0,7%. Кроме того, они подчеркивают, что после последовательного трехмесячного замедления экономического спада годовые темпы снижения ВВП в июле снова ускорились. Также в Минэкономразвития России добавляют, что негативная динамика ВВП сохраняется, прежде за все, за счет длительного сокращения объемов строительства и ограничений спроса.

Текст 10

Россия: экономика падает, поможет ли бюджету приватизация?

Российская экономика в первом квартале может потерять до 2,5%, считает Банк России. Ранее регулятор полагал, что падение не превысит 2%. Ухудшение прогнозов Центробанк объяснил негативной внешнеэкономической конъюнктурой. В прошлом году, как заявил накануне Росстат, ВВП сократился на 3,7%, в текущих ценах российская экономика оценивается в 80,4 трлн рублей (1 трл долларов). Нефть за минувший год подешевела вдвое. В этом году "внешнеэкономическая конъюнктура" пока не лучше: в январе стоимость барреля эталонной нефти Brent (на базе которой вычисляется цена российского экспортного сорта Urals) опускалась

ниже 28 долларов. Падение цен на нефть до уровня начала 2000-х создает риски для бюджета, почти половину поступлений которого формируют нефтегазовые доходы

ПРИЛОЖЕНИЕ 4

Визуализация работы алгоритма k-Means

1. Разбиение семи текстов на различные темы (классический алгоритм)

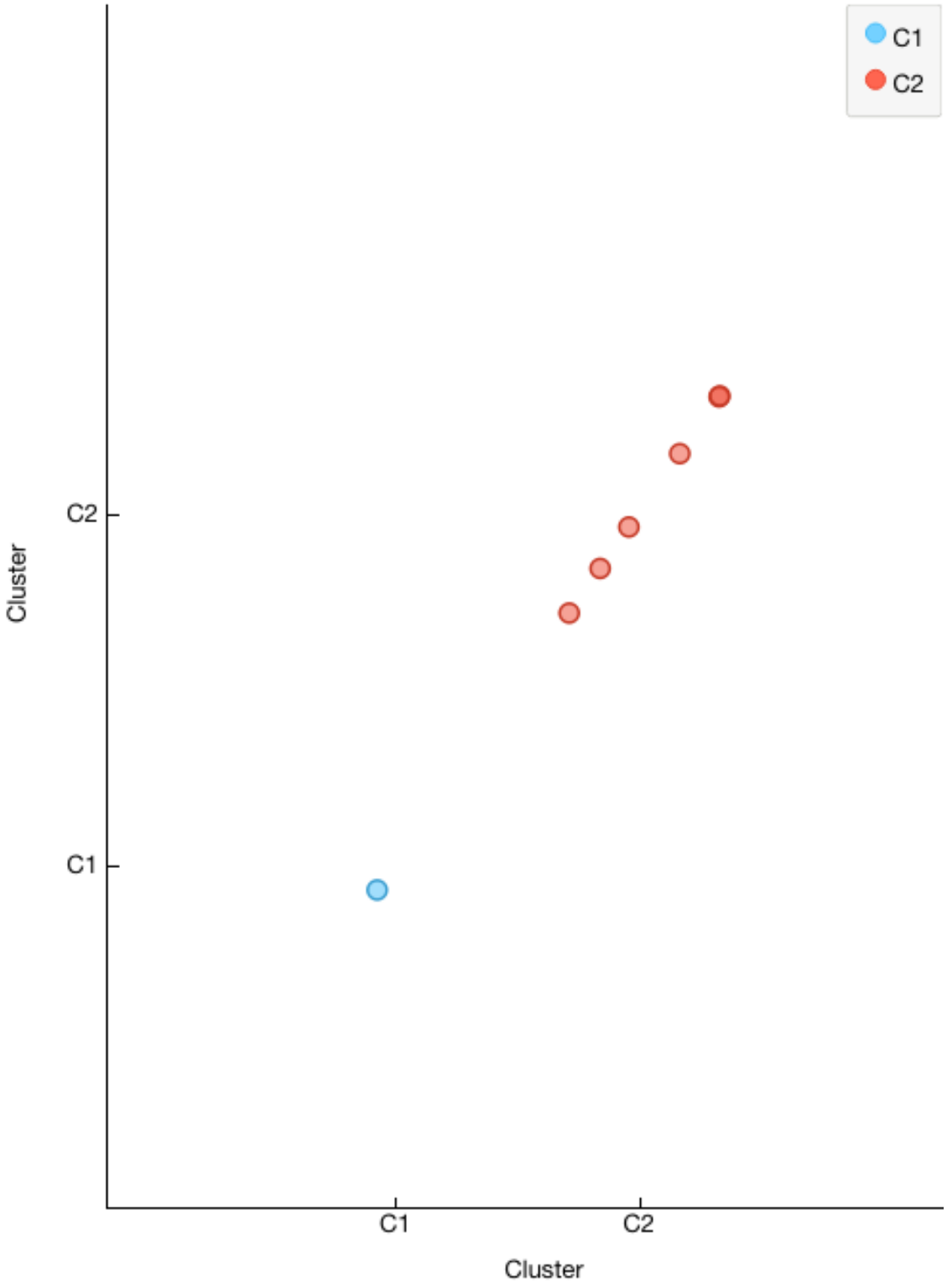


Рисунок П4.1 – Результат работы алгоритма KMeans для различных текстов

2. Разбиение семи текстов на различные темы (семантический алгоритм с использованием графов)

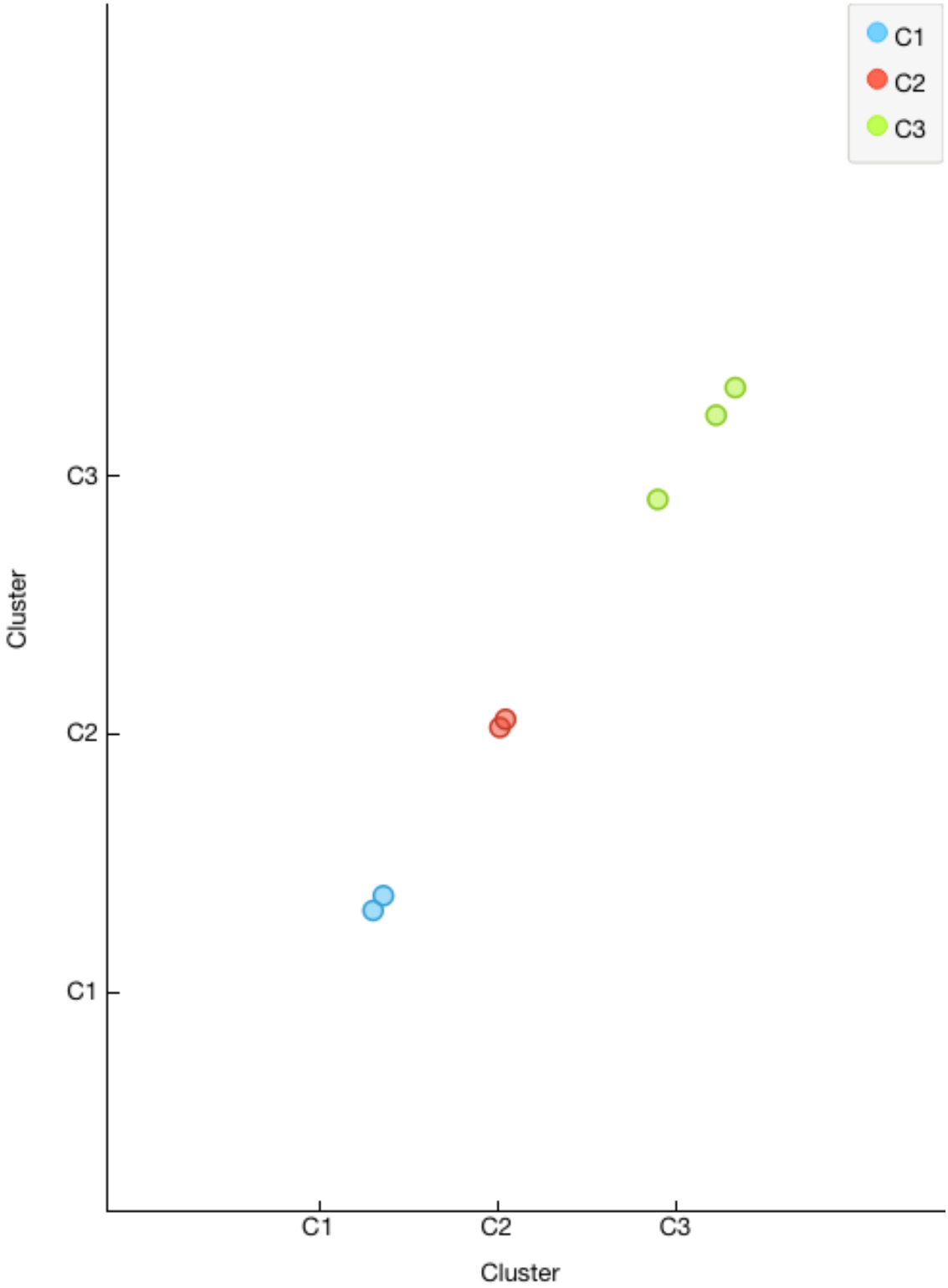


Рисунок П4.2 – Результат работы алгоритма KMeans для различных текстов (с учетом семантики)

3. Разбиение восьми текстов на схожие темы (классический алгоритм)

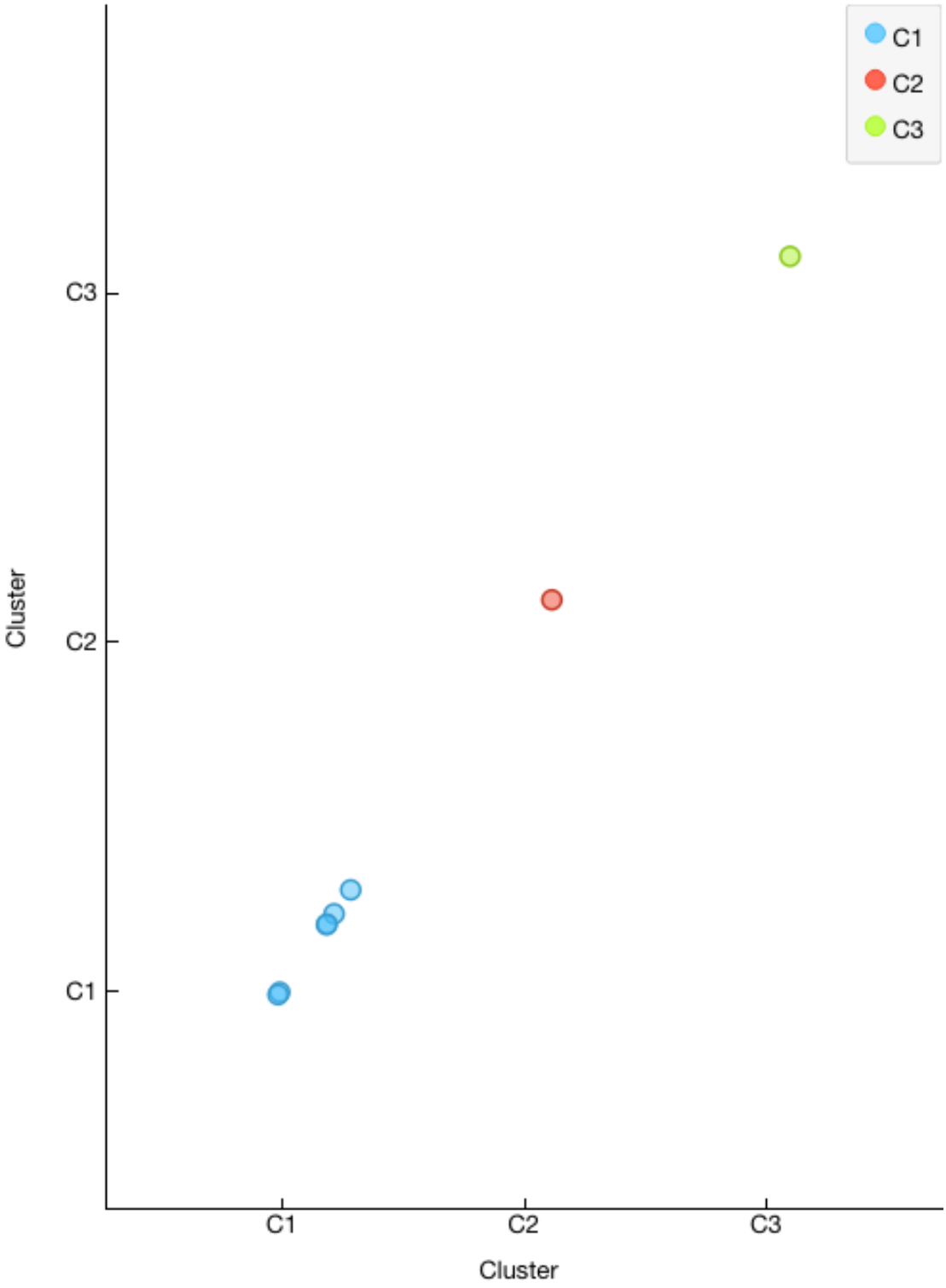


Рисунок П4.3 – Результат работы алгоритма KMeans для схожих текстов

4. Разбиение восьми текстов на схожие темы (семантический алгоритм с использованием графов)

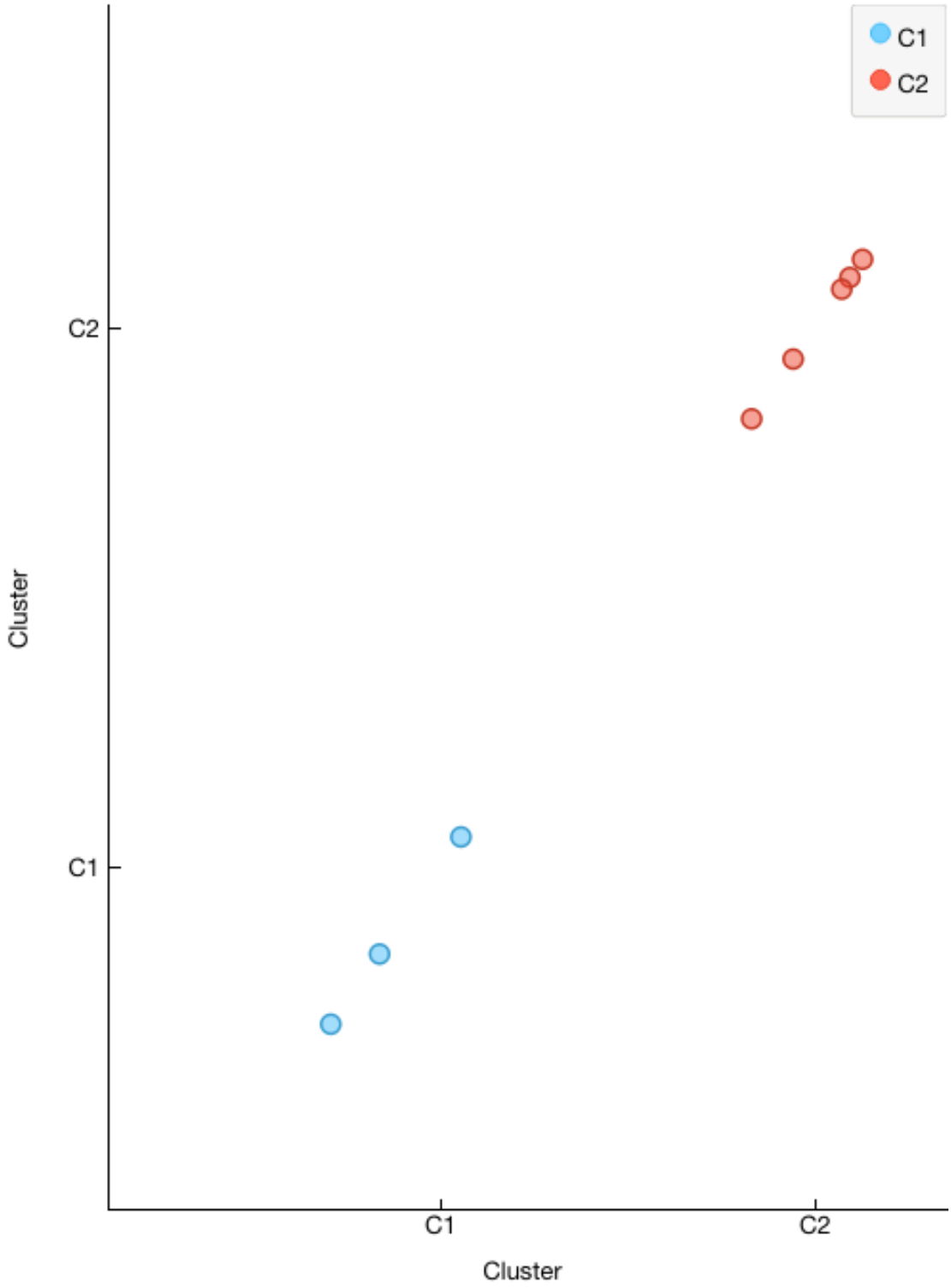


Рисунок П4.4 – Результат работы алгоритма KMeans для схожих текстов (с учетом семантики)

ЗАКЛЮЧИТЕЛЬНЫЙ ЛИСТ РАБОТЫ

Магистерская диссертация выполнена мною самостоятельно. Использованные в работе материалы и концепции из опубликованной научной литературы и других источников имеют ссылки на них.

Список использованных источников: 27 наименований.
Работа выполнена на 72 листах,
включая приложения на 12 листах.

Один экземпляр сдан на кафедру.

Подпись_____/_____/
(фамилия, инициалы)

Дата «_____» _____ 20_____г.