Математические структуры и моделирование 2017. № 3(43). С. 108–121

УДК 004.93
DOI: 10.25513/2222-8772.2017.3.108-121

ПРИМЕНЕНИЕ КЛАСТЕРНОГО АНАЛИЗА МЕТОДОМ К-СРЕДНИХ ДЛЯ КЛАССИФИКАЦИИ ТЕКСТОВ НАУЧНОЙ НАПРАВЛЕННОСТИ

Ю.А. Осипова
студент, e-mail: fray09@inbox.ru
Д.Н. Лавров
к.т.н., доцент, e-mail: dmitry.lavrov72@gmail.com

Омский государственный университет им. Ф.М. Достоевского

Аннотация. В статье рассматриваются вопросы построения программного обеспечения для выработки рекомендаций по назначению УДК. В основу разработки положен метод k-средних.

Ключевые слова: метод к-средних, лемматизация, TF-IDF.

Введение

Существует проблема определения принадлежности статьи к какой-либо из тематик или рубрик журнала и присвоения наиболее подходящего номера универсальной десятичной классификации (УДК). Очень часто авторы научных статей назначают УДК своей статье не на основе существующих справочников, а по примеру статьи схожей тематики. Тем не менее, тематика может немного меняться. Такой подход к назначению УДК после цепочки переприсвоений приводит к тому, что тема последней статьи может совсем не соответствовать назначенному УДК первой статьи. УДК у таких статей будет одинаковым. Решение простое — редакторы журнала должны проверять УДК, назначенные авторами статей.

Для автоматизации этого процесса необходимо разработать программное обеспечение классификации научных текстов для выработки рекомендаций по назначению первичного УДК научной статье.

Для достижения этой цели в работе решаются следующие задачи:

1. Провести сравнительный обзор существующих методов лемматизации текстов.
2. Расширить функционал программного обеспечения путём внедрения выбранных методов лемматизации.
3. Провести эксперименты для подтверждения работоспособности разработанного ПО.

1. Обзор существующих решений

1.1. Основные определения

Рассмотрим основные термины и определения, необходимые для дальнейшего изложения материала.

Лемма — основная или словарная форма слова. Например, для русского языка это именительный падеж единственного числа для существительного, инфинитив для глагола и т. д.

Лемматизация — это процесс, который использует морфологический анализ для приведения слова к лемме (нормальной форме). Также включает в себя определение лексической категории и применение различных правил нормализации для каждой части речи.

Лемматизатор — программный продукт для обработки естественного языка, который, используя правила лемматизации, возвращает начальную форму слова. Отдельную сложность составляет сама структура нашего языка. Русский язык преимущественно флективный, что означает и высокую его гибкость и аналитическую сложность. Особенность языков данного типа в том, что словоизменение происходит с помощью флексий — формантов, несущих в себе множество значений. Например, род, число и падеж для окончаний. Такая многовариантность обуславливает трудность определения леммы в большом количестве словоформ.

Определим критерии, на которых необходимо сосредоточить внимание при сравнении рассматриваемых методов:

1. Язык. Возможно ли использование данного решения для лемматизации текстов на русском языке.
2. Метод. То, каким образом метод приходит к результату.
3. Формат результата. В зависимости от метода меняются и возвращаемые данные: в одном случае — это основа слова, в другом — полная морфологическая информация.
4. Универсальность — может работать со всеми кодировками, типами данных и т. д.
5. Отдельное внимание стоит уделить тому, какую задачу решает данный метод.

Проанализируем существующие решения по заданным параметрам. Это позволит выбрать наиболее подходящую стратегию для решения поставленной задачи.

1.2. Стемминг

Стемминг — эвристический процесс, который находит основу для заданного слова. Основа не обязательно должна совпадать с морфологическим корнем слова.

Стемминг используется поисковыми системами для расширения запросов — query expansion (QE), а также является частью процессов лемматизации текстов на естественном языке. QE необходим для переформулировки запросов

для улучшения производительности в системах поиска информации, особенно в контексте понимания запросов.

Существует множество вариаций алгоритма стемминга. Один из наиболее подходящих для решения поставленной задачи — алгоритм усечения окончаний.

Данный алгоритм содержит список правил, который используется для нахождения основы слова с учётом его формы. Большинство из них выглядит следующим образом:

- если конец слова = «ый», обрезать «ый»;
- если конец слова = «ть», обрезать «ть»;
- если конец слова = «ому», обрезать «ому».

За счёт отсутствия справочных таблиц и полного перебора обеспечивается эффективность алгоритма по отношению к другим представителям семейства. Также за счёт настройки правил он работает с флективными языками. Однако это не означает, что полученный результат будет означать существование такого слова в лексиконе языка.

1.3. Stanford CoreNLP

Stanford CoreNLP — набор инструментов для анализа естественного языка. Он используется для получении форм слова, их частей речи; также для нормализации даты, времени и числовых величин; для разметки структуры предложения (указывает, какие существительные относятся к одним и тем же объектам).

Основные преимущества Stanford CoreNLP:

- быстрый и надёжный анализ текста;
- общая аналитика самого высокого качества;
- поддержка ряда основных естественных языков;
- доступные интерфейсы для большинства современных языков программирования;
- интегрированный инструментарий с хорошим набором инструментов грамматического анализа.

Stanford CoreNLP легко применяет множество инструментов лингвистического анализа к фрагменту текста. Код библиотеки представлен на языке Java и разрешён для использования в некоммерческих проектах под лицензией GPL.

С помощью настроек можно изменить, какие опции и инструменты должны быть включены, а какие отключены. CoreNLP объединяет многие инструменты: POS-тэгер, распознаватель названной сущности (NER), синтаксический анализатор (парсер), систему разрешения связей, анализатор тональности.

1.4. AOT

Проект AOT — набор инструментов, используемых для автоматической обработки текстов на естественном языке. В основном применяется для русского языка.

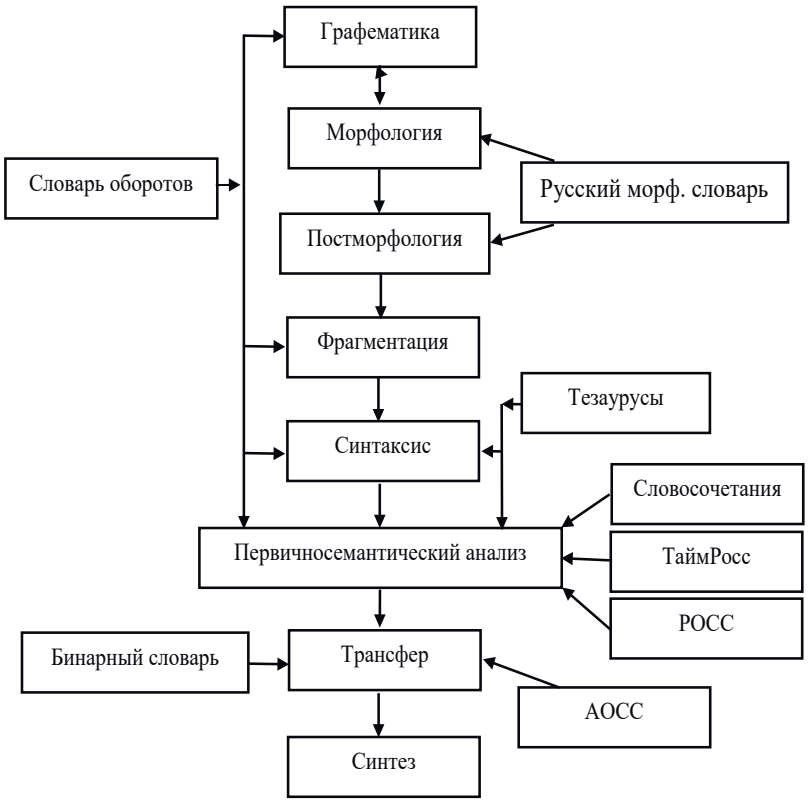

Рис. 1. Компоненты проекта АОТ

Суть данного механизма — в грамотной декомпиляции языковых механизмов, которая позволяет максимально приблизить человеческий язык к языку, понятному компьютеру.

Набор состоит из множества компонентов, которые один за другим обрабатывают исходный текст (см. рис. 1). Вход одного лингвистического процесса является выходом другого. Выделяются такие процессоры как графематический, морфологический, синтаксический и семантический [1]. Основан на грамматическом словаре А.А. Зализняка (1987). На данный момент содержит 161 тыс. лемм.

Для каждого слова входного текста формирует набор морфологических интерпретаций следующего типа:

- лемма;
- морфологическая часть речи;
- набор общих граммем (относящихся ко всевозможным словоформам парадигмы слова);
- многообразие наборов граммем.

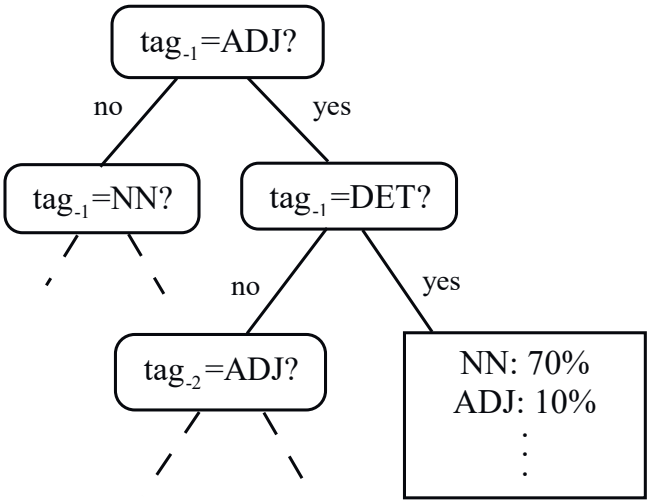

Рис. 2. Пример дерева решений Tree Tagger

1.5. Tree Tagger

Инструмент разработан Хелмутом Шмидом в университете Штутгарта. Это языконезависимый метод морфологической разметки текстов на естественном языке. Он основан на деревьях принятия решений и применяется в задачах обработки текстов на множестве языков. Исходный код данного метода закрыт. Распространяется в виде архива с исполняемыми файлами и набором языковых компонент.

Суть алгоритма в том, что оценка вероятности того, какой частью речи является слово, определяется бинарным деревом принятия решений. Вероятность определяется следующим образом: путь внутрь дерева считается до тех пор, пока лист не будет достигнут [2]. На ветвях дерева записаны условия, от которых зависит целевая функция, а на листьях — значения этой функции.

Рассмотрим пример на рис. 2, который демонстрирует образец дерева решений. Вероятность данной триграммы определяется следующим образом: продолжать процесс, пока операция не достигнет листа дерева. Например, если мы ищем, с какой вероятностью существительному предшествует определяющее слово и прилагательное, сперва мы должны ответить на вопрос в корневом узле. Если тег предыдущего слова — прилагательное, то выбираем положительную ветвь. Следующий ответ тоже положительный и, таким образом, алгоритм достигает концевого узла. Теперь остаётся лишь узнать вероятность тега n в таблице, прикреплённой к коду.

1.6. Сравнение рассмотренных решений

Были рассмотрены основные алгоритмы лемматизации текстов на естественном языке. В таблице 1 приведены общие сведения, необходимые для сравнения алгоритмов по параметрам, приведённым в начале обзора:

Таблица 1. Сравнение существующих алгоритмов лемматизации

Алгоритм	Русский язык	Метод	Формат результата	Универсальность	Задача
Алгоритм усечения окончаний	Есть	Стемминг	Основа слова	—	Поисковые системы
Stanford Core NLP	Нет	Словарный	Лемма	UTF-8	Анализ текстов
AOT	Есть	Словарный	Лемма + часть речи	UTF-8, только кириллица	Машинный перевод
Tree Tagger	Есть	Дерево принятия решений	Лемма + часть речи	—	Анализ и обработка текстов

- алгоритм — название алгоритма лемматизации;
- русский язык — возможность использования решения для текста на русском языке;
- метод — какой метод используется данным решением;
- формат — вид полученного результата в ходе работы системы;
- универсальность — особенности входных данных для текущего решения;
- задача — изначальное назначение данной системы.

Из таблицы видно, что нет явного лидера. Кроме того, неясна скорость работы этих методов. Поэтому было принято решение об использовании нескольких методов лемматизации для достижения поставленной цели.

2. Реализация

Задача разрабатываемого программного обеспечения — помощь в присвоении номера универсальной десятичной классификации. Результат работы программы должен быть представлен в виде списка предпочитаемых УДК со степенью соответствия УДК тексту статьи

Основные этапы работы разработанного ПО представлены на рис. 3.

Построение системы логически разбивается на две задачи: подготовка входных данных, необходимых для кластеризации, и собственно кластеризация.

2.1. Подготовка данных

2.1.1. Получение текстов

На этом этапе необходимо получить данные из заранее собранной базы текстов sqllite в формате: идентификационный номер, название, содержание, номер УДК.

База сформирована из 150 текстов, опубликованных в журнале «Математические структуры и моделирование» с 2010 года, у которых определён номер УДК редакторами.

Затем каждый документ передаётся в список и хранится там для дальнейшей обработки.

С помощью графического интерфейса (рис. 4) вводится текст, УДК которого необходимо узнать. Также определяются параметры, необходимые для дальнейшей работы программы.

2.1.2. Лемматизация

На данном этапе нормализуется каждое слово в текущем списке текстов. Для этого выбранному пользователем лемматизатору на вход подаётся содержание каждого текста из листа. Во время этого процесса отбрасываются стоп-слова, не имеющие смысловой нагрузки. Например, предлоги, местоимения, союзы, междометия и т. д.

Затем содержание каждого документа токенизируется и отдельные слова обрабатываются непосредственно алгоритмом, с учётом особенностей каждого лемматизатора. После чего из полученного списка слов формируется корпус слов всех текстов.

$$corp = \begin{bmatrix} w_1, \ldots, w_i \\ l_1, \ldots, \_i \end{bmatrix},$$

где w — слова, содержащиеся в корпусе, а l — лист документов, в которых оно содержится.

Для каждого элемента списка корпуса подсчитывается количество текстов, в которых он встречается.

По завершении этого процесса отсекаются слова с наименьшим употреблением для ускорения работы алгоритма и уменьшения вектора корпуса, используемого для каждого документа.

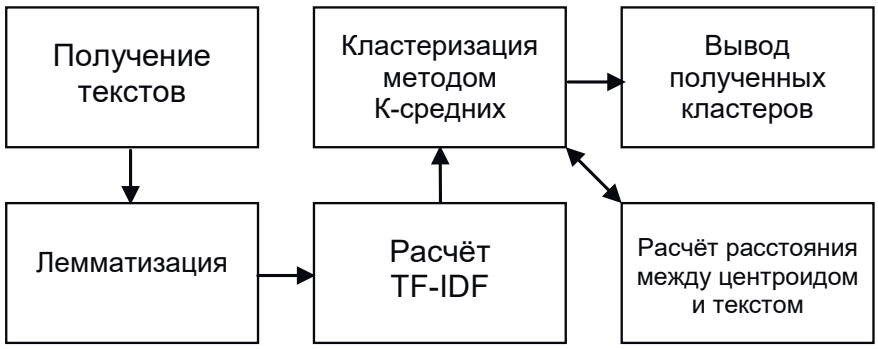

Рис. 3. Этапы работы разработанного ПО

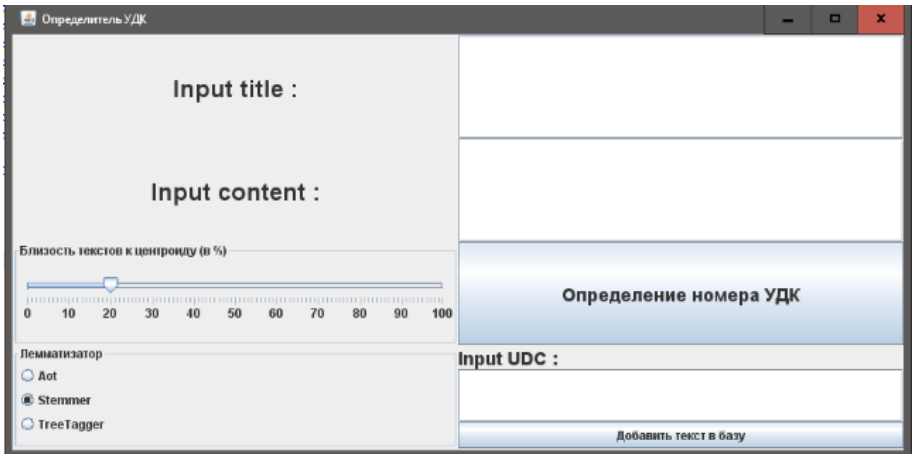

Рис. 4. Графический интерфейс

2.1.3. Расчет TF-IDF

С помощью кодировщика, использующего TF-IDF, содержание каждого документа кодируется в вектор. TF-IDF (от англ. TF — term frequency, IDF — inverse document frequency) — мера статистического анализа важности слова в контексте документа. Используется при расчёте близости текстов друг к другу. Состоит из двух частей TF и IDF.

TF (от англ. — частота слова) — численное значение вхождения заданного слова в текущем документе. Рассчитывается по формуле:

$$\mathrm{T}F = \frac{n_i}{\sum n_k},$$

где n_i — количество вхождений данного слова, n_k — общее число слов в документе.

IDF (от англ. — обратная частота документа) — численное значение, показывающее с какой частотой данное слово встречается во всех исходных документах. Расчётная формула:

$$IDF = \log\left(\frac{D}{d_i}\right),$$

где D — общее количество документов, а d_i — документы, в которых встречается данное слово.

Конечное значение коэффициента TFIDF равно произведению выше представленных множителей

$$TFIDF = TF \cdot IDF.$$

Больший вес получают слова с высокой частотой в пределах данного документа и с низкой частотой в пределах всего набора документов.

Для расчёта TF используется вектор нормализованных слов. Каждое слово становится ключом в карте, а количество вхождений — значением.

За основу IDF берётся вектор корпуса слов, полученный ранее. Его длина задаёт размерность вектора TF-IDF для каждого документа. Это необходимое условие при расчёте расстояния между двумя векторами.

Затем определяется метрика, необходимая для расчёта расстояния. Так как выбранный метод k-средних, то и метрика — евклидово расстояние, которое рассчитывается по формуле:

$$d = \sqrt{\sum_{i}^{n} \left(x_i - x'_i\right)^2},$$

где x и x′ — точки, между которыми рассчитывается расстояние.

На этом заканчивается подготовительный этап и начинает работу алгоритм кластеризации.

2.2. Кластеризация

После этого полученные данные передаются непосредственно алгоритму кластеризации. Именно он участвует в формировании групп сходных по смыслу текстов.

В качестве алгоритма кластеризации выбран метод k-средних.

Алгоритм k-средних — один из наиболее популярных и простых в реализации методов кластерного анализа. Это неиерархичный метод неконтролируемого обучения. Он позволяет разделить произвольный набор данных на заданное число кластеров так, что объекты внутри одного кластера были достаточно близки друг к другу, а объекты разных не пересекались. Другими словами, цель этого алгоритма — объединить в группы сходные данные по некоторым заданным критериям (см. рис. 4). Чаще всего при кластеризации используются меры расстояния. По умолчанию данный алгоритм использует квадрат Евклидова расстояния.

В 1950-х годах независимо друг от друга Гуго Штейнгауз и Стюарт Ллойд опубликовали свои работы [3, 4]. Однако особую популярность метод получил после работы Маккуина. Детальное описание и возможные изменения даны Джейном и Дабсом.

Алгоритм итеративен, то есть повторяется до тех пор, пока не будет выполнено условие выхода. Начнём с того, что на всём множестве доступных значений D выберем k центроидов (центров масс). Затем каждую точку из множества D привязываем к ближайшему центроиду. После того как все точки распределены, пересчитываем центры масс для каждого кластера. А дальше идёт повтор двух предыдущих действий до тех пор, пока центроиды не перестанут сдвигаться или действие алгоритма искусственно не прервётся.

К достоинствам данного алгоритма относятся следующие характеристики. Во-первых, результат работы алгоритма не зависит от порядка следования данных. Во-вторых, простота реализации.

Недостатки метода следующие. Первый — необходимость заранее задавать число кластеров. Но эту проблему достаточно просто решить. Одно из таких

возможных решений — повторить несколько раз алгоритм при предполагаемых значениях. И затем выбрать наилучшую кластеризацию. Также несколько решений этой проблемы представлены в книге [5].

В текущей реализации для решения проблемы с необходимостью предварительной передачи количества кластеров число кластеров k устанавливается в наименьшем значении, которое обеспечивает расстояние от внешнего кластера до внутреннего. Данное расстояние рассчитывается как отношение среднего расстояния между кластерами к среднему расстоянию между документами в кластере. Таким образом, нам необходимо задать только кластеризационный порог, относительно которого и будет строиться необходимое число кластеров. После чего на каждой итерации цикла будет рассчитано распределение документов для текущего числа кластеров до тех пор, пока расстояние между текущим количеством кластеров не станет ниже кластеризационного порога.

Также для более тонкой настройки пользователю даётся возможность выбрать расстояние, показывающее насколько близко к центру кластера должны располагаться вектора документов, чтобы считаться частью кластера.

Ниже приведены этапы работы алгоритма.

- Центроидом первого кластера становится TF-IDF вектор того документа, УДК которого необходимо определить. Заполняем остальные k кластеров достаточно удалёнными документами из общего набора. Их нормализованные вектора слов — начальные центроиды.
- Распределяем оставшиеся нераспределёнными документы в прилагаемом списке документов по ближайшим кластерам, предоставляемым набором кластеров.
- Обновляем центроиды кластеров в предоставляемом наборе кластеров.
- Повторяем шаги 2 и 3 до тех пор, пока число итераций не будет равно заданному выше кластеризационному порогу.

В результате работы данного алгоритма получаем кластеры со сходными по содержанию текстами. После чего пользователю предоставляются полученные данные в наглядной форме (см. рис. 5).

3. Эксперимент

3.1. Сравнение алгоритмов лемматизации

Для эксперимента по определению оптимального лемматизатора была собрана база из 150 текстов, опубликованных в журнале «Математические структуры и моделирование» с 2010 года, у которых определён номер УДК редакторами журнала. Затем был выбран текст того же журнала выпуском ранее 2010 года. Общее число слов равно 306215. Число слов, обработанных лемматизатором равно 198844. Число запусков равно 100.

Большая разница между исходным числом слов и слов, подающихся лемматизатору, обусловлена сложностью обработки формул. Большинство из них сформировано с помощью функций в LaTeX, поэтому они некорректно отображаются непосредственно в самой базе данных и целевом тексте. При предва-

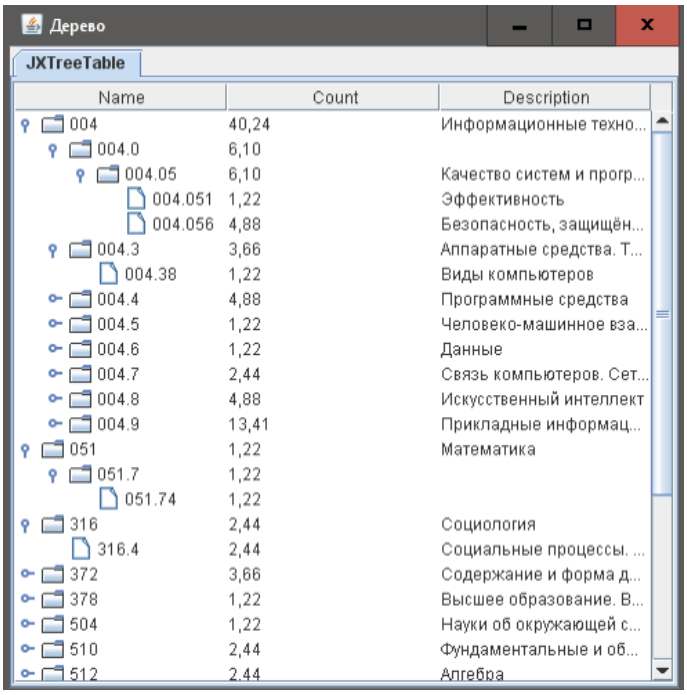

Рис. 5. Данные, предоставляемые пользователю

Таблица 2. Результаты работы на тестовых текстах

Название алгоритма	Среднее время выполнения (в сек.)	Скорость (слов/сек.)	Выходное число слов корпуса (-5% низко употребляемых)
Стеммер	2,48	80179	2020
Aот	55,42	3588	2006
Tree Tagger	759,59	262	2365

рительной обработке они отсекаются вместе со стоп-словами для корректного выполнения программы.

В таблице 2 собраны результаты на тестовом наборе текстов. Стеммер подтвердил свою скорость. Рекомендуется использовать в качестве лемматизатора Стеммер и АОТ, т. к. временные затраты на их использование существенно меньше.

Для повышения точности необходима более обширная база текстов, с несколькими представителями каждого номера УДК, используемого в журнале. Также есть вероятность ошибки в определении номера в уже обработанных текстах (см. рис. 4), что повышает вероятность погрешности.

3.2. Проверка работоспособности разработанного ПО

Для эксперимента по определению эффективности работы программы взята уже собранная база из предыдущего эксперимента и статьи на русском языке из ещё не опубликованного номера журнала «Математические структуры и моделирование».

Из 13 полученных текстов в 11 результат первого уровня УДК совпал с присвоенным редактором, что составляет 84,6% верно проведённых вычислений. Для дальнейшей оценки используем 11 оставшихся статей, в 6 верно определён номер универсальной десятичной классификации второго уровня, что составляет 55% корректной оценки.

Причиной таких результатов являются:

1. Использование формул и специальных символов, не воспринимаемых текстовым редактором.
2. Наличие типичных для данного журнала УДК, например, 510 и 514. Для рассматриваемого журнала характерны такие номера как 004, 517 и 519, а также их подуровни. Для решения этой проблемы необходимо расширить текстовую базу. Существуют номера журнала «Математические структуры и моделирование», вышедшие ранее 2010 года, однако, опубликованные в них статьи не проходили проверку редактором.
3. Также в одном из текстов изначально не был определён УДК ниже второго уровня, что не позволяет говорить о точности работы программного обеспечения (ПО) в отношении этой статьи.

В ходе экспериментального использования ПО была подтверждена эффективность данной разработки, так же выявлены проблемы текущего этапа разработки и обозначены пути их решения.

Заключение

Программная реализация метода k-средних подтвердила его эффективность, однако, подтвердилась и основная проблема при практическом использовании метода — сложность в выборе количества итераций. Слишком большое число повторений может привести к тому, что сложность алгоритма увеличится до $O(k^2)$. Она может быть решена практическим перебором необходимого числа итераций, а также предварительным анализом данных.

В ходе работы были достигнуты следующие результаты.

1. Проведён сравнительный обзор существующих методов лемматизации текстов.
2. Расширен функционал программного обеспечения путём внедрения выбранных методов лемматизации.
3. Проведены эксперименты, подтверждающие работоспособность разработанного ПО.

В качестве дальнейшего развития можно рассмотреть следующие направления.

Таблица 3. сводные данные по эффективности работы

N	Заголовок	Присвоенный номер УДК	Рекомендованный номер УДК
1	Моделирование и оптимизация процесса распределения человеческих ресурсов и аудиторного фонда вуза на основе анализа учебных планов	004.912+021	004.0
2	Применение метода построения ассоциативных правил к анализу деятельности общественных организаций	004.93	004.9
3	О редукции модальностей в деонтических исчислениях	510.643	519.2
4	Разработка системы мандатного управления доступом для операционных систем семейства Windows	004.42	004.4
5	Вопросы существования устойчивого цикла в одной модели молекулярного репрессилятора	514.745.82	517.9
6	Идентификация параметрической модели изменения численности зачисленных абитуриентов по малой выборке	004.000	004.9
7	Равновесная динамика лесных экосистем с учётом взаимосвязи «растительность-почва»	004.9:631.4 + 519.6	519.2
8	Аналитическое и численное представление звуковых характеристик обобщённой центрированной волны	519.6	519.7
9	Введение зависимости в схему испытаний Бернулли	519.2	519.2
10	Латентный анализ на базе метода штрафных функций для многомерных бинарных показателей	519.237.07	519.1
11	FRIS-компактность групп пациентов с артериальной гипертензией	519.237.07	519.7
12	Квантование массы и группа Лоренца	512.815.8	512.8

1. Расширение и пополнение базы текстов. А именно — постоянные пополнения базы данных из обновляющейся библиотеки, например, elibrary.ru или CyberLeninka с полноценной серверной частью, работающей без вмешательства пользователя.
2. Построение более высокоуровневой модели «обучения с учителем» на основе полученных данных от метода k-средних.

Разработанное программное обеспечение передано в редакцию журнала «Математические структуры и моделирование» для использования в работе редакции и сбора отклика пользователей с целью дальнейшего совершенствования ПО.

Благодарности

Выражаем признательность Богаченко Надежде Фёдоровне и Вишняковой Ольге Анатольевне за внимание, проявленное к нашей работе.

Литература

1. Sokirko A. A short description of Dialing Project. 2001. URL: `http://www.aot.ru/docs/sokirko/sokirko-candid-eng.html` (дата обращения: 28.08.2017).
2. Schmid H. Probabilistic Part-of-Speech Tagging Using Decision Trees. URL: `http://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger/data/tree-tagger1.pdf` (дата обращения: 28.08.2017).
3. Steinhaus H. Sur la division des corps materiels en parties. Bull. Acad. Polon. Sci., 1956. P. 801–804.
4. Lloyd S. Least square quantization in PCM's. Bell Telephone Laboratories Paper, 1957. URL: `http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.131.1338&rep=rep1&type=pdf` (дата обращения: 28.08.2017).
5. Gordon A.D. Classification, 2nd Edition. CRC Press, 1999. 272 p.

APPLICATION OF CLUSTER ANALYSIS BY THE K-MEANS METHOD FOR THE CLASSIFICATION OF SCIENTIFIC TEXTS

U.A. Osipova
Student, e-mail: fray09@inbox.ru
D.N. Lavrov
Ph.D. (Eng.), Associate Professor, e-mail: dmitry.lavrov72@gmail.com

Dostoevsky Omsk State University

Abstract. The article discusses the construction of software to develop recommendations for the appointment of UDC. The development is based on the k-means method.

Keywords: method of k-means, lemmatization, TF-IDF.

Дата поступления в редакцию: 29.06.2017