Московский государственный университет имени М.В. Ломоносова
Факультет вычислительной математики и кибернетики
Кафедра системного программирования

Сидоренко Юрий Анатольевич

Автоматическое извлечение ключевых понятий из текста с учетом иерархической структуры предметной области

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

Научный руководитель:
д.ф-м.н., профессор
В. А. Серебряков

Москва, 2016

СОДЕРЖАНИЕ

1 ВВЕДЕНИЕ

Текст – есть один из самых удобных способов сохранения и накопления информации в современном мире. Объем данных, накопленных людьми в текстовом формате, уже превысил тот объем, который один человек может осмыслить и структурировать за всю свою жизнь, и он продолжает расти все более быстрыми темпами так, что все больше и больше усилий уходит на ручную структуризацию этих данных. В связи с этим в последние десятилетия начали появляться различные подходы с целью автоматической или полуавтоматической структуризации накопленных знаний в текстовом формате. Одним из таких подходов является выделение ключевых понятий из текста. Этот подход и рассматривается в данной работе.

Любой текст, будь то книга, научная статья, или пост в интернете, может быть рассмотрен как набор ключевых понятий и отношений между ними. Отношения между ключевыми понятиями, на самом деле, так же могут быть представлены как ключевые понятия, которые присутствуют в данном тексте. Человек-читатель текстового источника информации, как правило, четко для себя выделяет понятия, о которых идет речь в тексте, а также он их ранжирует по степени важности. Если у такого читателя спросить «О чем данный текст?», в его рассказе будут фигурировать те самые ключевые понятия, которые он посчитал наиболее важными для данного текста.

Таким образом, ключевые понятия в тексте несут в себе довольно большую часть информации, которую текст содержит. По таким понятиям человек может приблизительно восстановить темы, присутствующие в тексте, связи между этими темами, понятия, которые интересуют его больше всего, например, личности или географические локации.

Исходя из этого, выделение ключевых понятий из текста активно применяется в следующих областях:

- Тегирование документов, статей, постов в интернете для их структурирования, кластеризации и поиска по ним.
- Выделение наиболее важных частей текста, автоматическое реферирование.
- Обнаружение упоминаний интересующих нас понятий в тексте.
- В других задачах обработки текстов.

Задача выделения ключевых понятий из текста может рассматриваться в разных постановках в зависимости от наличия тех или иных данных о предметной области в виде корпуса текстов по предметной области, тезауруса и других ресурсов [1, Pp. 16-21]. Такие данные позволяют построить приближение модели, в рамках которой мыслит человек, когда он читает текст.

2 Постановка задачи

2.1 Разновидности задачи

Есть несколько разновидностей задачи выделения ключевых понятий из текста. Они отличаются, в основном, по следующим критериям:

- текстовый информационный источник, из которого выделяются ключевые понятия;
- источник терминологии, из которого берутся слова для описания ключевых понятий;
- количество выделяемых ключевых понятий.

2.1.1 Текстовый информационный источник

Одна характеристика, по которой задача выделения ключевых понятий из текста может приобретать тот или иной акцент, – это текстовый информационный источник. Этим источником, как правило, может быть либо один документ, либо целый корпус документов.

Если в качестве информационного источника выступает один документ, то задача требует выделения тех ключевых понятий из документа, которые наиболее точно передают содержание этого документа.

Если в качестве информационного источника выступает коллекция документов (корпус), то эта задача представляет собой задачу выделения терминологии из корпуса текстов.

В данной работе рассматривается именно первый тип задачи – выделение ключевых понятий из предоставленного документа. Потенциально, второй тип задачи – выделение терминологии – может быть сведен к первому типу – выделение ключевых понятий из документа, хотя это не всегда удобно и, обычно, задачу выделения терминологии на корпусе текстов рассматривают как обособленную задачу обработки текстов.

2.1.2 Источник терминологии

Можно выделить три основных варианта представления ключевых понятий, а именно, для представления каждого ключевого понятия:

- используются термины из фиксированного словаря;
- используются лишь те термины, которые встречаются в предоставленном документе или предоставленном корпусе, на котором рассчитываются статистические величины, такие как частота встречаемости термина;
- используются произвольные термины, не обязательно присутствующие среди терминов корпуса, предоставленного алгоритму.

Если существует возможность заранее зафиксировать словарь, например, используя тезаурус предметной области, то задача значительно упрощается в силу того, что необходимо выделять лишь термины, присутствующие в заданном словаре. Это так же позволяет не беспокоиться о проблемах производительности, связанных с ростом словаря n-gram на больших объемах текста в модели выделения ключевых понятий в виде n-gram. Однако, этот подход имеет серьезный недостаток, а именно, далеко не для всех предметных областей существует полный и точный словарь терминов предметной области, что ведет к невозможности применить данный подход. Более того, такой словарь может устаревать со временем по мере поступления в базу знаний новых документов, содержащих термины ранее не встречающиеся.

Второй подход более гибок в этом плане. Он использует термины, которые встречаются в документе (либо в коллекции документов) в качестве ключевых понятий, что позволяет расширять словарь терминов со временем в случае необходимости. В данной работе используется именно этот подход.

Третий подход – это подход, который используется людьми: человек способен выделить по термину в тексте саму суть понятия, которое описывает этот термин, и в некоторых случаях пометить это понятие еще более подходящим термином. Однако, такой подход, хоть и является наиболее близким к человеческому решению проблемы, не получил на данный момент

широкого распространения в силу недостатка информации о том, как следует строить модель предметной области, которая будет надежно представлять понятия этой предметной области вместе с их связями.

2.1.3 Количество ключевых понятий

Количество выделяемых ключевых понятий – еще одна характеристика, по которой можно выявить несколько различных задач. Эта характеристика, как правило, определяется прикладной задачей, в рамках которой рассматривается сама задача выделения ключевых понятий из текста.

Количество может быть сравнительно небольшим, например, до десятка ключевых понятий. Прикладные задачи, которые используют такое количество ключевых понятий, - это, в основном задачи, в которых происходит присвоение категорий небольшим статьям, интернет постам. Примером может быть принадлежность вопроса на сайте Stackoverflow.com одной из заранее определенных категорий.

Несколько большее количество ключевых понятий документа, например, от десяти до ста, может использоваться в задачах автоматического реферирования или задачах структуризации текста, таких как тематическое моделирование и кластеризация корпуса документов по темам.

Наибольшее количество ключевых понятий выделяется на целом корпусе текстов в рамках задачи выделения терминологии предметной области на основе корпуса текстов на темы этой предметной области [2].

В данной работе приоритет отдается выделению небольшого и умеренного количества ключевых понятий – первой и второй категориям, описанным выше. Хотя никаких искусственных ограничений на количество ключевых понятий не устанавливается.

2.2 Ключевые понятия глазами машины

Допустим, у нас есть документ. Как определить, какие понятия в нем ключевые? Человек определяет ключевые понятия на основе модели предметной области, которую он сформировал в процессе получения своего жизненного опыта. Нам необходимо создать аналогичную модель, близкую той

модели, которой пользуется человек. Чтобы создать такую модель, можно использовать корпус текстов. На этом корпусе будут рассчитаны некоторые статистические значения, которые лягут в основу нашей модели. Более того, хотелось бы выделять ключевые понятия в терминах интересующих нас предметных областей, поэтому корпус должен представлять собой документы, относящиеся к одной или нескольким предметным областям.

Многие современные методы выделения ключевых понятий из текста не берут в расчет принадлежность документа к предметной области. В результате, ключевые понятия, которые выделяются, используя такие подходы, получаются не терминами какой-либо предметной области из предоставленного корпуса, а уникальными, редко встречающимися терминами предоставленного корпуса. Такие алгоритмы хорошо подходят для Named-entity recognition [3] – задачи выделения личностей, организаций, локаций, и т. д. Однако если нас больше интересуют понятия, которые несут определенный смысл в рамках одной из заданных предоставленным корпусом предметных областей, такие подходы работают плохо. Они не предназначены для этой задачи. В связи с этим возникает необходимость в другом подходе, который позволяет выделять ключевые понятия, принадлежащие к одной из предметных областей. Именно такой подход я исследую в своей работе.

2.3 Формулировка задачи

Итак, в связи с предпосылками, описанными выше, возникает следующая задача:

- На входе имеется корпус документов. Предполагается, что каждый документ относится к одной или нескольким предметным областям. Т.е. часть документов корпуса формирует тему (или, эквивалентно, предметную область). Это разбиение может быть либо задано вручную человеком, либо его можно произвести на основе автоматической кластеризации или тематического моделирования.
- На входе так же имеется документ, из которого предполагается выделять ключевые понятия.

- На выходе нужно получить ранжированный по релевантности набор слов и словосочетаний, каждое из которых представляет ключевое понятие из текста документа, данного на входе.

3 Обзор существующих решений рассматриваемой задачи или ее модификаций

Задача выделения ключевых понятий из текста изучается уже давно [1, Pp. 6-8]. В рамках данной работы были рассмотрены и изучены основные существующие подходы к решению этой проблемы. Среди них наиболее популярные – Kea [4], Kea++ [5], Maui [6], DBpedia Spotlight [7]. Как правило, алгоритмы, лежащие в основе таких решений, существенно не зависят от конкретного естественного языка и могут быть перенесены с одного языка на другой без значительных изменений.

3.1 Используемые признаки

Для выделения ключевых понятий из текста каждый из таких алгоритмов подсчитывает некоторые статистические значения на корпусе текстов. Для Maui в качестве корпуса берется набор статей на определенную тему. DBpedia Spotlight использует статьи ресурса Wikipedia в качестве своего корпуса, и, в дополнение к нему, RDF хранилище DBpedia - для извлечения дополнительной информации о связях между ключевыми понятиями [8].

Среди статистических значений, которые рассчитываются на основе предоставленного корпуса, присутствуют:

- Document Frequency (сокращенно DF) термина – частота встречаемости термина в документах предоставленного корпуса текстов;
- Inverse Document Frequency (сокращенно IDF) термина – инвертированное значение Document Frequency термина;
- Term Frequency (сокращенно TF) термина – частота встречаемости термина в документе, из которого требуется выделить ключевые понятия.

Иногда используются и менее очевидные признаки, например, позиция, на которой в предложениях встречается термин, количество слов, из которых состоит термин, признаки на основе Wikipedia, и другие признаки, на которых не имеет особого смысла заострять внимание.

Одним из признаков на основе Wikipedia является признак Keyphraseness – способность кандидата в ключевые понятия быть ключевой фразой. Этот признак позволяет отдавать предпочтение кандидатам, используемым на Wikipedia в качестве ссылок на другие страницы, т.е. это кандидаты, которые были помечены авторами статей Wikipedia как ключевые понятия. Авторы продукта Maui используют данный признак [1, Pp. 94-97] как одно из улучшений над продуктами Kea и Kea++.

3.2 Функция оценки релевантности термина

Далее, на основе выделенных признаков для каждого термина, который является кандидатом в ключевые понятия, определяется оценка на его релевантность в данном документе. Как правило, эта оценка представляет собой функцию, отображающую пространство признаков в действительное значение. Чем оценка термина получится выше, тем более важным понятием этот термин является в предоставленном документе.

Таким образом, такая функция оценки приближает модель человеческого восприятия ключевых понятий.

В некоторых работах, например, в продукте DBpedia Spotlight, такая функция подбирается вручную людьми на основе эвристических предположений [6, Pp. 50-61].

В рамках продуктов Kea, Kea++ и Maui авторы рассматривают возможность применения методов машинного обучения для определения функции оценки кандидатов в ключевые понятия [1, Pp. 37-41]. Авторы сравнивают алгоритм наивной байесовской классификации кандидатов в ключевые понятия с алгоритмом классификации на основе решающих деревьев. При сравнении результатов, полученных на основе эвристик, с результатами, которые дает машинное обучение, становится понятно, что в определенных случаях алгоритм машинного обучения способен значительно лучше определить форму функции оценки на основе размеченных данных, чем предложенные человеком эвристики.

Однако, серьезной проблемой методов, использующих машинное обучение, является необходимость предоставления размеченной тренировочной выборки, которая, как правило, представляет собой корпус документов, где в каждом документе выделены ключевые понятия. Помимо трудозатрат на создание такой тренировочной выборки еще одним недостатком такого алгоритма является его привязанность к предметной области, на которой алгоритм был обучен. Так как, каждая предметная область имеет свою модель, которая наилучшим образом учитывает особенности этой предметной области, алгоритм придется переобучать.

3.3 Особенность признака Inverse Document Frequency

Все вышеперечисленные алгоритмы используют в качестве своего основного признака для выделения ключевых понятий признак Inverse Document Frequency.

Кандидаты, выделенные на основе IDF признака – это кандидаты, которые встречаются в очень небольшом количестве документов корпуса. То есть IDF позволяет выделять уникальных в предоставленном корпусе кандидатов. Такие кандидаты почти никогда не являются терминами предметных областей предоставленного корпуса.

Для некоторых задач хотелось бы выделять кандидатов, которые являются терминами одной из предметных областей предоставленного корпуса. Для этого можно использовать признак Document Frequency (DF). Проблемой такого подхода является то, что среди кандидатов с высоким значением DF будет много стоп-слов (таких слов, которые не несут значительной информации). Избавиться от них можно либо вручную, используя список стоп-слов, либо автоматически.

Вручную отсеивать стоп-слова довольно сложно, потому что они уникальны в каждой предметной области. Например, помимо союзов, предлогов, и подобных часто встречающихся слов, слово «функция» может рассматриваться как стоп-слово в корпусе документов на тему «Математика». В то время как корпус на тему «Биология» имеет совершенно другие стоп-слова.

Предлагаемый в данный работе подход позволяет избавиться от стоп-слов автоматически за счет использования принадлежности документов корпуса некоторым темам.

4 Исследование и построение решения задачи

Введем основные определения и обозначения, которые будут использоваться в дальнейшем при описании предлагаемого в данной работе подхода.

4.1 Основные определения и обозначения

Ключевое понятие для документа – слово или словосочетание, которое наиболее точно отражает содержание этого документа.

Кандидат для документа – слово или словосочетание из документа.

Корпус – коллекция документов.

Тема в корпусе – набор понятий, для которых принято считать, что они имеют более крепкие связи между собой, чем с другими понятиями.

Предметная область в корпусе – альтернативное название темы в корпусе.

Кластер в корпусе – подмножество документов корпуса, которые относятся к одной теме. Т.е. кластер привязывает конкретный документ к конкретной теме, присутствующей в корпусе.

Кандидат для корпуса – слово или словосочетание, которое является кандидатом для хотя бы одного документа корпуса.

$$\begin{aligned}&\text{DF(кандидата) на корпусе}\\&= \text{Количество документов корпуса, которые содержат кандидата}\\&\div \text{Количество документов в корпусе}\end{aligned} \quad (1)$$

$$\text{IDF(кандидата) на корпусе} = 1 \div \text{DF(кандидата)} \quad (2)$$

$$\begin{aligned}&\text{TF(кандидата) для документа}\\&= \text{количество раз, которое кандидат встретился в документе}\end{aligned} \quad (3)$$

Термин для корпуса – кандидат, который на этом корпусе имеет $DF \geq A$, где A – некоторая константа, определяемая экспериментально.

Стоп-слово на корпусе – кандидат корпуса, не являющийся информативным ни для какой предметной области корпуса.

4.2 Подход к извлечению ключевых понятий на основе иерархической структуры предметной области

В данной работе рассматривается подход, который учитывает принадлежность документа к одной или нескольким предметным областям. Этот подход позволяет автоматически избавиться от стоп-слов, присваивая им оценки, значительно ниже оценок, которые даются терминам текста, несущим значительную информационную нагрузку.

В данной секции описываются общие идеи предлагаемого подхода. Детали его реализации будут уточняться в практической части. Сразу следует отметить, что сам подход не зависит существенно от какого-либо конкретного естественного языка.

4.2.1 Основная идея подхода, использующего иерархическую структуру предметной области

Пусть предоставленный на входе корпус разбит на темы. То есть каждый документ принадлежит одной или более темам. В такой постановке корпус текстов представляет собой набор тем, причем, эти темы могут пересекаться по документам, в которых они присутствуют.

Стоп-слова – это такие термины, которые не несут информационной нагрузки по отношению к темам предоставленного нам корпуса. Если термин встречается лишь в документах одной предметной области, то есть основания считать, что такой термин несет некоторую информационную нагрузку по отношению к этой предметной области. Если же термин размазан по темам, т.е. он присутствует в документах сразу нескольких предметных областей, то это довольно надежный признак для того, чтобы считать, что такой термин не несет информационной нагрузки по отношению к какой-то одной предметной области.

Этот факт позволяет избавиться от стоп-слов автоматически, если отслеживать принадлежность кандидата в ключевые понятия кластерам предоставленного корпуса.

4.2.2 Определение уникальности термина

Имея набор предметных областей в корпусе, логично предположить, что предметная область, в документах которой термин встречается наиболее часто, и является истинной предметной областью этого термина. В связи с этим введем понятие наиболее близкого кластера для кандидата.

Наиболее близкий кластер для кандидата – это кластер с максимальным DF этого кандидата. Предметная область такого кластера потенциально наиболее близка кандидату документа.

Для того чтобы определить, на сколько кандидат уникален среди всех кластеров предоставленного корпуса, введем понятие уникальности кандидата в одном кластере относительно другого кластера – Relative Cluster Uniqueness(RCU).

$$\begin{aligned}\text{RCU(кандитата) в cluster\_1 по отношению к cluster\_2} \\ = (\text{DF(кандидата) в cluster\_1} \div \text{DF(кандидата) в cluster\_2}) \\ \times (\text{DF(кандидата) в cluster\_1} - \text{DF(кандидата) в cluster\_2})\end{aligned} \quad (4)$$

Это позволит нам сравнивать уникальность кандидата в его наиболее близком кластере по отношению ко всем другим кластерам. Если посмотреть на RCU более внимательно, то можно заметить, что RCU получается на основе двух составляющих: отношения значений DF кандидата в двух кластерах и разности значений DF этого кандидата в тех же кластерах.

Отношение значений DF кандидата – это основной признак относительной уникальности кандидата в одном кластере по отношению к другому. Чем оно выше, тем больше уверенность, что кандидат является термином одного кластера и при этом не является термином другого. Однако, одно лишь отношение значений DF кандидата не позволяет различать между маленькими и большими абсолютными значениями DF кандидата. А делать

такое различие хотелось бы в силу того, что более предпочтительны те кандидаты, которые являются терминами предметных областей корпуса, а не кандидаты, которые имеют низкие значения DF, чувствительные к шуму в корпусе. В связи с этим вводится поправочный признак – разность значений DF кандидата.

Разность значений DF кандидата – это вспомогательный признак относительной уникальности кандидата в одном кластере по отношению к другому. Он позволяет оценочной функции отдать предпочтение тем кандидатам, про которых можно с уверенностью сказать, что они являются терминами одной из предметных областей предоставленного корпуса.

Теперь легко определить основной признак, по которому можно будет сказать, в какой степени хорошим является кандидат в ключевые понятия с точки зрения принадлежности его темам корпуса. Этот признак назовем кластерной уникальностью – Cluster Uniqueness(CU).

Для определения Cluster Uniqueness кандидата выберем наиболее близкий кластер для этого кандидата и посчитаем Relative Cluster Uniqueness кандидата в этом кластере по отношению ко всем другим кластерам. Тогда, просуммировав все подсчитанные RCU, получим:

$$\text{CU(кандидата) на кластеризованном корпусе} = \text{Сумма RCU(кандидата)} \quad (5)$$

Данная сумма показывает, в какой степени кандидат уникален к своему наиболее близкому кластеру. В этой работе я использую это понятие для ранжирования кандидатов в ключевые понятия по релевантности. Функция CU определяется как сумма из соображений простоты и интуитивного удобства. Она показывает довольно хорошие результаты на практике, хотя и, может быть, заменена на какую-нибудь другую функцию, лучше принимающую в расчет особенности прикладной задачи.

4.2.3 Ранжирование кандидатов

Для того, чтобы ранжировать кандидатов предоставленного документа по их релевантности по отношению к данному документу, я использую два признака, определенных выше:

- Cluster Uniqueness(CU)
- Term Frequency(TF)

Cluster Uniqueness позволяет отдавать предпочтение кандидатам, т.к. в них выше уверенность в том, что они являются терминами лишь одной предметной области предоставленного корпуса.

Term Frequency позволяет выделять кандидатов, которые в предоставленном на входе документе упоминаются чаще других, а значит, более релевантны в контексте этого документа.

Для ранжирования кандидатов по релевантности определим ранг кандидата – Rank:

$$\mathrm{Rank}(\text{кандидата}) = \mathrm{TF}(\text{кандидата}) \times \mathrm{CU}(\text{кандидата}) \qquad (6)$$

Каждый выделенный из документа кандидат получает значение Rank. Именно Rank и используется для финального ранжирования кандидатов в ключевые понятия. Посчитанный таким образом ранг кандидата в ключевые понятия учитывает в себе, в первую очередь, уникальность этого кандидата среди всех тем корпуса и использует частоту встречаемости кандидата в документе как поправку, способную различать кандидатов с близкими значениями CU по релевантности.

Следует отметить, что стоп-слова, даже несмотря на высокие значения TF, получают очень низкое значение Rank в силу распределенности по темам корпуса.

5 Описание практической части

Предлагаемый в данной работе подход к выделению ключевых понятий был реализован в качестве программного продукта.

Язык программирования Python был выбран в качестве языка реализации по той причине, что на данный момент он поддерживает большое количество удобных библиотек и инструментов для обработки текстов.

5.1 Алгоритм выделения ключевых понятий

Алгоритм выделения ключевых понятий состоит из двух этапов:

- подсчет статистических величин в документах предоставленного корпуса;
- выделение ключевых понятий из документа.

5.1.1 Подсчет статистических величин в документах предоставленного корпуса

В программе указывается путь к корпусу. Текущая реализация программы предполагает, что директория, в которой расположены документы корпуса, устроена следующим образом:

- в самой директории присутствует набор папок;
- каждая папка содержит документы, относящиеся к общей, разделяемой между документами, теме.

Если документ содержит более одной темы, он может быть реплицирован в несколько папок с соответствующими темами.

Далее для каждой директории, содержащей документы одной темы, создается объект, который хранит статистику признаков кандидатов из данной темы. Для каждой темы, считываются документы друг за другом. Каждый документ программа преобразует в набор кандидатов. По мере чтения документов и обнаружения новых кандидатов, программа обновляет объект, в котором хранится статистика кандидатов документов определенной темы. Как только все документы были считаны, этот этап завершается.

Преобразование документа в набор кандидатов происходит следующим образом:

- из документа выделяются токены - слова длинной больше двух букв;
- для этих слов используется стемминг - для приведения похожих слов к общей форме;
- на основе токенов генерируются всевозможные последовательные n-grams длинной от одного до трех.

Получившиеся в результате такой обработки n-grams используются в качестве кандидатов в ключевые понятия.

Среди статистических величин, которые рассчитываются на корпусе, присутствуют TF(кандидата) в каждом документе и количество документов, в которых присутствует каждый кандидат, на основе чего в дальнейшем получается DF(кандидата) на кластере каждой из тем. Эти значения DF(кандидата) в дальнейшем используются для расчета значений RCU(кандидата) на одном кластере относительно другого кластера, которые, в свою очередь, являются компонентами признака CU(кандидата).

В итоге, по окончании этого этапа, за один проход по всему корпусу собирается вся необходимая статистика о кандидатах, по которой для каждого кандидата можно подсчитать CU за константное время.

5.1.2 Выделение ключевых понятий из документа

После того, как вся статистика на предоставленном корпусе была собрана (в текущей реализации эта статистика хранится в оперативной памяти), алгоритм готов к этапу, ради чего все и затевалось, – этапу выделения ключевых понятий. Программе предоставляется документ. Этот документ программа считывает и преобразует в набор кандидатов теми же средствами, которыми преобразование документа в набор кандидатов происходило на этапе подсчета статистики.

TF каждого кандидата рассчитывается для каждого документа в процессе преобразования документа в набор кандидатов. В силу того, что основной

составляющей в формуле ранга кандидата является Cluster Uniqueness, а Term Frequency является больше вспомогательным признаком, я сглаживаю эффект TF, беря от TF логарифм по основанию два.

В результате, для каждого кандидата в предоставленном документе считается Rank по формуле, описанной ранее. Кандидаты сортируются по убыванию ранга и выводятся пользователю в виде списка.

5.2 Архитектура приложения

Реализованное приложение содержит три основных класса:

- Document
- Corpus
- ClusteredCorpus

Взаимодействия между этими классами находятся в модуле под названием Main.py, который так же является и точкой входа в программу. Далее следует описание обязанностей каждого из основных классов.

5.2.1 Класс Document

Основной задачей данного класса является преобразование документа в набор кандидатов. В процессе преобразования объект этого класса так же собирает статистику встречаемости частот кандидатов в документе, которую он предоставляет через свой интерфейс объектам других классов.

У каждого кандидата есть свой представитель – это форма кандидата, которую он имеет до нормализации (в данном случае, до стемминга). Несколько представителей кандидата в документе могут быть свернуты в одного кандидата и дальше все действия производятся с нормализованными кандидатами. Однако, пользователю приложения ключевые понятия хотелось бы представлять в виде фраз, присутствующих в предоставленном документе, а не в виде некой нормализованной формы. Поэтому возникает подзадача для каждого кандидата нормализованной формы – найти лучшего представителя этого кандидата. В данной работе я использую в качестве представителя форму, наиболее часто встречающуюся в предоставленном корпусе.

Для этого каждый документ для каждого кандидата хранит список представителей этого кандидата, который в дальнейшем используется классом Corpus для получения представителя, наиболее часто встречающегося во всем корпусе.

5.2.2 Класс Corpus

Основной задачей класса Corpus является подсчет DF каждого кандидата на корпусе документов. Данный класс «оборачивает» список объектов класса Document, и за один проход по этому списку он собирает статистику значений частот, с которыми кандидаты, присутствующие в документах предоставленного списка, встречаются в этих документах. Вместе с частотами для каждого кандидата так же определяется его представитель, про которого речь шла выше.

Через свой интерфейс данный класс позволяет получить, среди прочей информации, значение DF для кандидата на корпусе.

5.2.3 Класс ClusteredCorpus

Как было описано ранее, на входе у этого алгоритма имеется кластеризованный корпус. Класс ClusteredCorpus представляет собой как раз абстракцию этого уровня. Он состоит из множества кластеров, каждый из которых представлен объектом класса Corpus. Каждый такой кластер представляет предметную область (или тему) в предоставленном алгоритму на входе корпусе текстов.

Основной задачей этого класса является подсчет значений Cluster Uniqueness для каждого кандидата, присутствующего хотя бы в одном из кластеров. Расчет CU каждого кандидата происходит за один проход по всем кандидатам. При этом, для каждого кандидата находится его наиболее близкий кластер и далее используется формула для определения CU, как описано ранее.

Через свой интерфейс данный класс позволяет получить значение CU для кандидата на кластеризованном корпусе.

5.3 Тестирование приложения

Для того чтобы протестировать работу написанного приложения, был собран корпус новостей на английском языке.

Этот корпус содержит 3 темы:

- Arms Control (2300 документов)
- Environmental Issues (3800 документов)
- Food and Agriculture Issues (1500 документов)

Средний размер документа в этом корпусе ~10 000 символов.

Был так же реализован базовый алгоритм выделения ключевых понятий на основе признаков TF и IDF. Ранг в этом алгоритме определяется по формуле:

$$\text{Rank}(\text{кандидата}) = \text{TF}(\text{кандидата}) \times \text{IDF}(\text{кандидата}) \quad (7)$$

Из этого корпуса были выбраны несколько документов, на которых производилась настройка и отладка реализованного в данной работе алгоритма. Также на этих документах запускался алгоритм на основе TF и IDF признаков и производилось сравнение этих двух алгоритмов.

Для документа из корпуса Arms Control сравнение между ключевыми понятиями, выделенными каждым из этих алгоритмов, выглядит следующим образом:

Cluster Uniqueness	**TF & IDF**
military	FRC
war	Hizbullah
armed	Israel
peace	PLO

forces	Sharon
killed	IDF
security	Lebanon
plant	war
bomb	Hamas
Netanyahu	Gaza
Israel	political-strategic
ceasefire	dismantle
Ehud	withstood
Islamic	Peres

Рис. 1 - Сравнение результатов работы алгоритмов

На рисунке 1 изображены лучшие кандидаты в ключевые понятия.

В первой колонке – по результатам алгоритма, предложенного в данной работе. Во второй – по результатам алгоритма на основе TF и IDF признаков. Всего выделенных из документа кандидатов в ключевые понятия значительно больше. В целях демонстрации было взято 14 кандидатов.

Не трудно заметить, что алгоритм, выделяющий ключевые понятия на основе IDF признака, стремится выделить очень специфические названия – понятия, которые являются редкими в предоставленном корпусе.

Алгоритм, предложенный в данной работе, напротив, стремится выделить ключевые понятия, несущие существенную информационную нагрузку в отношении одной из тем предоставленного корпуса – т.е. эти понятия являются терминами одной из предметных областей данного корпуса.

Таким образом, результаты, которые эти подходы дают, существенно отличаются, и каждый из них предназначен для решения своего множества задач.

6 ЗАКЛЮЧЕНИЕ

В данной работе исследуется проблема выделения ключевых понятий из текста, а так же изучаются особенности применения разных алгоритмов ее решения на практике.

Был предложен новый альтернативный существующим методам подход к извлечению ключевых понятий из текста. Этот подход был так же реализован в виде программного продукта, который позволяет применять его на разных корпусах текстов. Как показали результаты тестирования, в некоторых задачах обработки текстов предпочтение данному подходу может быть отдано вполне оправданно. Реализованный в этой работе метод позволяет автоматически избавиться от стоп-слов и выделить те ключевые понятия, которые несут информационную нагрузку в рамках одной из тем предоставленного на входе корпуса, иначе говоря, выделить ключевые понятия в терминах тем этого корпуса. Как и многие другие подходы к решению исследуемой задачи, предложенный в данной работе метод не накладывает никаких явных зависимостей на естественный язык, с которым он работает.

Как было отмечено выше, в некоторых ситуациях алгоритмы машинного обучения могут давать более качественную функцию оценки релевантности кандидата в ключевые понятия. Поэтому одним из возможных направлений дальнейшего исследования может являться применение таких алгоритмов в тандеме с предлагаемым в данной работе подходом для общего улучшения результатов в задаче выделения ключевых понятий.

СПИСОК ЦИТИРУЕМОЙ ЛИТЕРАТУРЫ

[1] Medelyan O. Human-competitive automatic topic indexing // The University of Waikato. – 2009.

[2] Астраханцев Н.А. Методы и программные средства извлечения терминов из коллекции текстовых документов предметной области // Федеральное государственное бюджетное учреждение науки Институт системного программирования Российской академии наук. – 2015. – С. 10-22.

[3] Nadeau D., Sekine S. A survey of named entity recognition and classification // National Research Council Canada / New York University. – 2007. – Pp. 1-4.

[4] Witten, I. H., G. W. Paynter, E. Frank, C. Gutwin, and C. G. Nevill-Manning. Kea: Practical automatic keyphrase extraction // In Proc. ACM Conf. on Digital Libraries, Berkeley, CA, US. New York, NY: ACM Press. – 1999. – Pp. 1-9.

[5] Medelyan O. Automatic keyphrase indexing with a domain-specific thesaurus // The University of Freiburg. – 2005. – Pp. 51-60.

[6] Maui [Электронный ресурс] // URL: http://www.medelyan.com/software (дата обращения: 06.05.2016).

[7] DBpedia Spotlight [Электронный ресурс] // URL: https://www.github.com/dbpedia-spotlight (дата обращения: 06.05.2016).

[8] Mendes P.N. Adaptive Semantic Annotation of Entity and Concept Mentions in Text // Wright State University. – 2013. – Pp. 36-42.

ПРИЛОЖЕНИЕ А

Кластеризация на основе ключевых понятий

Предложенный в данной работе подход основывается на том, что каждый документ корпуса, предоставленного на входе, отнесен к одной или нескольким темам, т.е. корпус должен быть кластеризован.

Стандартные алгоритмы выделения ключевых понятий на основе TF и IDF, как правило, не пользуются разбиением корпуса по темам и, соответственно, не предъявляют этого дополнительного требования. Это требование, на самом деле, не является таким же сильным, как, например, наличие репрезентативной тренировочной выборки для алгоритма, использующего машинное обучение.

Для того чтобы использовать специфику разбиения корпуса по темам при выделении ключевых понятий на произвольном корпусе, можно, например, воспользоваться:

- алгоритмами тематического моделирования;
- алгоритмами кластеризации корпуса документов.

Тематическое моделирование позволяет построить модель коллекции текстовых документов, которая определяет, к каким темам относится каждый из документов.

Кластеризация коллекции документов предполагает разбиение этой коллекции, как правило, на множество документов, не пересекающиеся между собой.

Эти алгоритмы, даже если они приведут к не совсем точному результату в сравнении со специалистом-человеком, позволят хотя бы примерно наметить границы предметных областей, чего уже будет достаточно для полноценного использования алгоритма выделения ключевых понятий, описываемого в данной работе.

Рассмотрим один из способов применения выделенных ключевых понятий на практике.

Пусть, у нас есть корпус научных статей, в котором каждая статья относится к одной из тем этого корпуса. Предполагается, что темы этого корпуса могут быть выстроены в древовидную иерархию, в которой листья результирующего дерева будут представлять наиболее узкие темы; эти темы могут быть объединены в общую тему на более высоком уровне иерархии, а корень, являясь самым высоким уровнем иерархии, будет представлять собой тему всего предоставленного корпуса. Нам необходимо найти такую иерархию. Она может выглядеть так, как показано на рисунке 2.

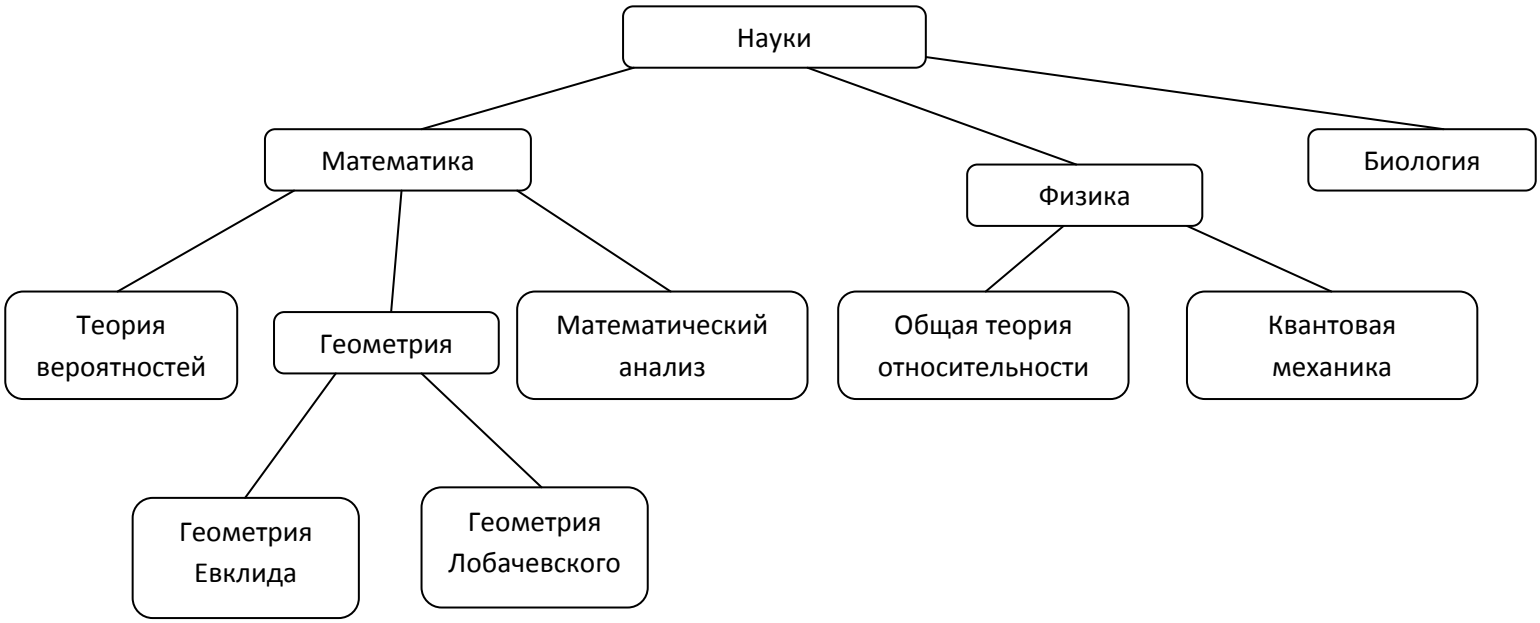

Рис. 2 - Корпус текстов, представленный в виде иерархической структуры предметных областей

Иерархию предоставленного корпуса мы будем строить по следующему алгоритму:

1. кластеризуем корпус, используя в качестве признаков документа всех кандидатов из корпуса;
2. для каждого документа корпуса выделяем ключевые понятия;
3. кластеризуем корпус, используя в качестве признаков документа ключевые понятия;

4. для каждого полученного на этапе 3 кластера повторяем шаги 1-3 и получаем очередной уровень в иерархии.

Количество уровней в иерархии, а так же количество кластеров на каждом уровне, может быть подобрано вручную, либо автоматически.

Данный алгоритм рассматривает на каждой итерации кластер, как корпус документов, который разбивается на несколько кластеров более низкого уровня. Сначала алгоритм рассматривает предоставленный корпус как кластер, который надо разбить на более мелкие части. Затем он последовательно применяется к каждой части и разбивает ее на еще более мелкие части.

Как отмечалось выше, первый этап этого алгоритма нужен для того, чтобы можно было выделить ключевые понятия в документах по алгоритму, предлагаемому в этой работе. Затем эти ключевые понятия используются для более качественной кластеризации корпуса, который рассматривается на текущей итерации.

В результате получается искомая древовидная иерархическая структура.