ВЫДЕЛЕНИЕ КЛЮЧЕВЫХ СЛОВ ТЕКСТА НА СТОРОНЕ КЛИЕНТА

С. Д. Кулик

Белорусский государственный университет, г. Минск;
sergey.d.kulik@gmail.com;
науч. рук. – В. М. Котов, д-р физ.-мат. наук, проф.

Выделение ключевых слов в текстах является актуальной и исследуемой в данный момент задачей. Оно позволяет грамотно организовать категоризацию документов, а также улучшить качество поиска информации.

В данный момент большое внимание уделяется алгоритмам на основе рекуррентных нейронных сетей; большой уклон сделан на обработку на стороне сервера. Но вместе с этим существуют и некоторые трудности: обработка на стороне третьих сервисов может быть дорогой, а в случае работы с конфиденциальными данными, может быть и вовсе недопустимой. Более того, на данный момент многие компании предоставляют интерфейс программирования приложений для выделения ключевых слов в текстах только на нескольких языках.

Выделение ключевых слов на стороне клиента призвано решить эти проблемы и найти разумный компромисс между точностью, простотой использования и поддержкой множества языков.

В данной работе будет предложен алгоритм для выделения ключевых слов на стороне клиента и проведен его сравнительный анализ с известными аналогами на корпусе, собранном из 27 тысяч заметок англоязычного ресурса WordPress [1].

Ключевые слова: анализ текстов; выделение ключевых слов; машинное обучение; вычисления на стороне клиента.

Ключевым объектом проводимого исследования являются ключевые слова. Формализуем и дадим определение этому понятию.

Пусть рассматривается некоторая коллекция документов, в которой текст каждого документа написан на каком-то из естественных языков. Выберем некоторый произвольный документ из этой коллекции. Ключевыми словами данного документа будем считать одно или несколько слов, или словосочетаний, характеризующих данный текст в выбранной коллекции. В широком смысле, ключевые слова документа можно рассматривать как его категории в данной коллекции.

Задача проводимого исследования – разработка и анализ эвристических алгоритмов для выделения ключевых слов документов веб-страниц на стороне клиента. Ограничение о выполнении кода на клиентской стороне накладывает некоторые ограничения на класс применяемых алгоритмов, так как это означает, что полученные алгоритмы должны быстро и эффективно работать на обычных персональных электронно-вычислительных машинах, в распоряжении пользователей которых нет кластеров с сотнями или тысячами процессоров. При этом, полученные

алгоритмы не должны использовать дополнительных вычислительных мощностей в облачных и подобных сервисах.

Самой близкой по тематике к рассматриваемой работе является работа американских исследователей “Automatic keyword extraction from individual documents” [2]. Работа довольно широко известна как простой эвристический алгоритм для получения ключевых фраз документа и существует несколько реализаций приведенного в ней алгоритма на различных языках программирования, в том числе таких как Java Script и Type Script, использующимися в разработке веб-страниц на стороне клиента. Существует также библиотека, основанная на данном алгоритме на языке программирования Python, которая даже доступна в официальном репозитории пакетов языка PyPI [3]. Примечательна обозначенная работа также тем, что там приводится относительно эффективный, единый для всех языков метод выделения стоп-слов в языке рассматриваемого корпуса документов.

Несколько другой подход, применимый к анализу данных на клиентской стороне предлагается в работе “TextRank: Bringing Order into Texts” [4] исследователей Рады Михалцеа и Пола Тарау. В работе предлагается рассмотреть текст как неориентированный граф, в котором строится соответствие между вершинами и словами текста, ребрами отображаются связи между словами. Далее предлагается модифицированная версия алгоритма вычисления ранга Пейджа (алгоритм предложен в 1998 году Л. Пейджем и прочими в работе “The PageRank Citation Ranking: Bringing Order to the Web” [5]). Ранг Пейджа был изначально предназначен для решения задач из области информационного поиска, ключевым же моментом работы Михалцеа и Тарау является адаптация предложенного Пейджем и Брином инструментария для задачи анализа текстов.

Предлагаемый альтернативный подход заключается в построении схожим образом с работой [2] множества кандидатов в ключевые слова и фразы и их последующем ранжировании, состоящем в выделении множества ключевых особенностей каждого из кандидатов и вычислении ранга кандидата - суммы произведений количественного значения каждой из особенностей на ее вес.

Приведем способ построения обозначенного множества кандидатов. Как было установлено при анализе данных блоговых ресурсов, средняя длина указываемой человеком ключевой фразы составляет 1.28 слова, поэтому в нашем алгоритме при построении множества кандидатов будем рассматривать все различные комбинации из не более трех последовательно встречающихся слов анализируемого текста, не содержащих между собой знаков пунктуации и стоп-слов. Стоит отметить, что существует множество подходов к выделению стоп-слов или шумовых слов.

На практике, однако, часто используются их заранее заданные списки такие, как например список стоп-слов Фокса (Fox stoplist в англоязычной литературе), который был получен Кристофером Фоксом как результат работы “A Stop List for General Text” [6] в 1989 году.

Теперь приведем примеры признаков-факторов, используемых для ранжирования множества кандидатов, построение которого мы описали в предыдущем абзаце:

- Факт вхождения и количество вхождений кандидата в заголовок текста;
- Количество вхождений слова- или фразы-кандидата в анализируемый текст;
- Позиция первого вхождения рассматриваемого кандидата в рассматриваемый текст, пропорционально которой назначается штраф;
- Количество вхождений в текст отдельных слов рассматриваемой фразы-кандидата;
- Характеристики вершин графа, построенного по рассматриваемому тексту.

Для вычисления весов данных факторов был сначала использован метод поиска по решетке параметров. Данный метод получал на вход множество факторов и диапазон весов для каждого из них, а также обучающую выборку, содержащую 5 % от всех текстов, полученных для исследования. Далее перебирались все возможные значения весов для каждого из факторов; для каждой полученной комбинации было проведено построение множества кандидатов, ранжирование кандидатов из данного множества относительно друг друга, получены ключевые слова всех текстов выборки и вычислено значение метрики качества, равное числу ключевых слов и ключевых фраз, совпавших с элементами из эталонного множества ключевых слов и фраз для соответствующих текстов. В конце перебора параметров рассматриваемого алгоритма были выбраны те, при которых значение рассматриваемой метрики качества было наибольшим.

Далее данный подход был пересмотрен. Задача была сведена к задаче машинного обучения о бинарной классификации. В полученной задаче каждый кандидат в множество ключевых слов рассматривался как вектор значений его факторов. Каждый из векторов-примеров имел один из двух классов: класс принадлежности соответствующего ему кандидата ко множеству ключевых слов и класс, соответствующий отсутствию данной принадлежности. Отметим несбалансированность в рассматриваемых классах: так, текст из тысячи слов может потенциально иметь множество из трех тысяч кандидатов в ключевые слова и фразы; в то же время в этом множестве будет, чаще всего, не более десяти представите-

лей первого класса и несколько тысяч представителей второго. Таким образом, классификатор, предсказывающий всегда второй класс имел очень высокий показатель точности, что привело к необходимости применения мер по уравниванию рассматриваемых классов. Для этого, в качестве метрики качества в рассматриваемой задаче была выбрана метрика “площадь под кривой ошибок”, также классам были назначены веса таким образом, чтобы суммарный вес примеров первого класса и суммарный вес примеров второго класса совпали.

Для решения полученной задачи были обучены классификаторы на основании следующих алгоритмов машинного обучения: метод ближайшего соседа, метод опорных векторов, метод случайного леса, логистическая регрессия. Наиболее высокий результат на контрольной выборке показал метод логистической регрессии, обеспечивший площадь под кривой ошибок равную 0.861. Отметим также тот факт, что в силу того, что логистическая регрессия является линейной моделью, полученные в ней веса признаков могут быть использованы без изменений в качестве весов ранее построенных факторов.

Для оценки качества алгоритмов был реализован простейший поисковый робот для сбора корпуса текстов ресурса блогов на английском языке WordPress [1]. Данный робот позволил создать выборки для обучения и тестирования, суммарно состоящие из 27 тысяч статей данного ресурса. Так как каждая из статей имела в качестве атрибутов ключевые слова и фразы, стало возможным аннотировать тексты для проведения обучения и оценки качества.

В результате запуска на тестовой части выборки были получены следующие результаты: алгоритмом RAKE было правильно определено 5629 ключевых слов и фраз, алгоритмом TextRank было правильно определено 18615 ключевых слов и фраз, альтернативным же подходом, рассмотренным в данном исследовании, было правильно определено 39081 ключевых слов и фраз. Таким образом, при возможности добавления предобучения на корпусе связанных предметной областью или же языком в целом текстов на естественном языке, построенный метод может быть использован и показывать результаты, превосходящие аналоги без преодобучения. Использование данного метода на стороне клиента также не вызывает трудностей, так как сериализация модели состоит только лишь в переносе полученного в результате обучения вектора весов.

Библиографические ссылки

1. WordPress [Electronic resource]. URL: https://www.wordpress.com. (date of access: 26.05.2019).

2. Automatic keyword extraction from individual documents [Electronic resource] / S. Rose [et al.]. URL: https://www.researchgate.net/publication/227988510_Automatic_Keyword_Extraction_from_Individual_Documents (date of access: 26.05.2019).
3. PyPI – the Python Package Index. URL: https://www.pypi.org (date of access: 26.05.2019).
4. TextRank: Bringing Order into Texts [Electronic resource] / R. Mihalcea, P. Tarau. URL: https://www.researchgate.net/publication/200044196_TextRank_Bringing_Order_into_Texts (date of access: 26.05.2019).
5. The PageRank Citation Ranking: Bringing Order to the Web [Electronic resource] / L. Page [et al.]. URL: https://www.researchgate.net/publication/221996591_The_PageRank_Citation_Ranking_Bringing_Order_to_the_Web (date of access: 26.05.2019).
6. *Fox C.* A Stop List for General Text [Electronic resource]. URL: https://dl.acm.org/citation.cfm?id=378888 (date of access: 26.05.2019).