УДК 004.89

А.И. Парамонов, доц., канд. техн. наук (БГТУ, г. Минск)

МЕТОД ИНДЕКСИРОВАНИЯ ТЕКСТОВЫХ ФРАГМЕНТОВ ДЛЯ ОРГАНИЗАЦИИ СМЫСЛОВОГО ПОИСКА ПО БАЗЕ ДОКУМЕНТОВ

В настоящее время большая часть информации в бизнесе, промышленности, государственных и других учреждениях хранится в текстовой форме в базах данных, и эти текстовые базы данных содержат зачастую слабоструктурированные данные. Любой потенциально важный документ также может содержать некоторые в значительной степени неструктурированные текстовые компоненты. С каждым годом значительно увеличивающееся количество создаваемых цифровых документов и закономерное увеличение объёмов хранимых документов требует новых решений целого ряда задач. Эти задачи возникают в процессе управления данными и стадий, которые проходят данные от сбора и первичной обработки, сохранения и обеспечения доступа к ним, и до последующего использования и обработки с целью извлечения из них информации. Хорошая организация документооборота предполагает оперативный доступ к требуемым документам и организовать их своевременную обработку.

Проблема поиска информации в больших массивах текстовой информации актуальна для всех видов деятельности. Для её решения были разработаны алгоритмы от простейшего поиска по ключу (прямое нахождение слов запроса в документах) до построения моделей документов [1] и последующего сравнения их с моделью запроса. В работе рассматривается решение задачи индексирования документов для организации смыслового поиска по документной базе с ранжированием результатов и сопоставлением документов.

Для решения задачи индексации базы документов известны различные методы, основанные на специальных структурах данных, в числе которых инвертированный индекс, суффиксные деревья, матрица термов в документах и другие. Инвертированный индекс хранит общую информацию обо всей документной базе. Данная структура эффективно применяется для поиска документов, в которых встречается вхождение одно или несколько заданных слов [2]. Но сами документы не содержит сжатых представлений, что делает этот метод не пригодным для решения задачи поиска схожих документов. Суффиксные деревья [3] эффективны для поиска документов, содержащих не только определенное слово, а в принципе любую подстроку (любой

размерности). При этом операция сравнения документов реализуется очень трудоемко и выполняется с большими временными затратами.

Матрица термов в документах [4] для каждого документа содержит информацию о входящих в него термах (словах или словосочетаниях). Основной проблемой такого представления является «разреженность» матрицы. В связи с этим, чаще всего она храниться в структуре подобной инвертированному индексу («прямой индекс»), которая для каждого документа хранит список входящих в него слов и количество вхождений. Такая структура требует меньше памяти. Документ в данном индексе сжимается до списка входящих в него слов и их количества, или до вектора по универсуму терминов документной базы. Такие вектора являются отображением документа в некоторое векторное пространство. Сравнение векторов документов можно оценивать похожесть документов по содержанию слов. Одной из проблем использования «прямого индекса» для организации представления документа заключается в различном распределении отдельных слов в языке. Например, предлоги и многие глагольные формы встречаются в тексте намного чаще, чем более специфичные термы, которые тем не менее лучше отражают смысл текстового документа. Эта проблема решается путем модификации матрицы термов. В ячейку записывается не сам терм, а значение Tf-Idf [4,5] для данного терма в конкретном тексте и даже в наборе текстов. Такая модификация позволяет взвесить компоненты вектора документа так, что менее важные для определения содержания документа термы имеют меньшее значение в векторе, а значит и меньше влияют на результат сравнения текстов.

Следует отметить, что один из главных недостатков рассмотренных подходов к поиску по базе документов заключается в том, что никоим образом не учитывается структура текста. Терм может часто встречаться в тексте и быть «маркером» для некоторого документа, то есть выделять его из всего набора текстов, но при этом вообще не отражать смысл самого документа [6]. В качестве результата операции индексирования должно получаться некоторое представление документа, которое будет отражать его основное содержание. В качестве такого представления документа предлагается использовать вектор по универсуму терминов документной базы. Терминами выступают извлеченные из документов ключевые слова и фразы.

Извлечение ключевых слов предлагается выполнять с помощью метода поиска ключевых слов TextRank [7]. Принцип его работы заключается в построении графа со словами в вершинах и подсчете на его основе для каждого слова числового рейтинга. Он показывает, насколько слово часто встречается в этом тексте в разных контекстах.

Чем больше это число, тем вероятнее, что слово важно для текста и отражает суть его содержания. Основной плюс алгоритма в том, что для извлечения ключевых слов он использует только статистики слов в текущем документе. Это значит, что он не требует хранения никакой дополнительной информации. В качестве результата операции индексирования должно получаться некоторое представление документа, которое будет отражать его основное содержание. В общем весь процесс индексирования в итоге состоит из предобработки текста, извлечения ключевых слов, формирования векторов для документов и последующего построения k-d дерева.

Для процесса ранжирования определен алгоритм вычисления меры схожести текстовых фрагментов. Вектора проиндексированных текстов помещаются в k-мерное дерево, так как данная структура данных обеспечивает быстрый поиск и ранжирование по набору k-мерных векторов. Для задачи ранжирования в качестве меры схожести запроса и документа предлагается использовать метрику Минковского с p=1 (расстояние городских кварталов, L1 норма), что соответствует количеству терминов, относящихся только к запросу и только к документу. Таким образом, тексты с меньшим значением расстояния более похожи на запрос по смыслу. При добавлении новых текстов k-мерное дерево разбалансируется. В этой связи, при накоплении определенного количества новых документов, необходимо перестраивать дерево в ходе работы метода. Вопрос частоты балансировки дерева требует дополнительного исследования и будет рассмотрен в дальнейшем.

Был проведен компьютерный эксперимент с предложенным методом, в рамках которого разработано программное средство в виде Python библиотеки. Библиотека включает несколько основных классов, которые являются интерфейсом для её пользователей и реализуют основные этапы рассмотренного метода. Обработка данных внутри библиотеки, включая предобработку текста, разбиение его на токены, построение графа, вычисление весов и прочие операции, реализовано с помощью функций в специальных файлах с открытым доступом для пользователей. В программной реализации использованы внешние Python библиотеки в открытом доступе: pymorphy2 (для определения частей речи), scikit-learn (для реализации структуры данных k-d дерево), summa (отдельные функции PageRank и TextRank), pymongo (для работы с сервером документно-ориентированной СУБД MongoDB).

В качестве базы текстов для тестирования был использован дамп базы данных русскоязычной википедии. Статьи википедии вполне соответствуют понятию документа, данному в начале обзора предметной области. Для улучшения качества результатов работы ал-

горитма к тексту применяется предобработка. Во-первых, из текста убирается пунктуация и служебные слова, которые сами по себе не несут смысловой нагрузки (предлоги, местоимения и прочее). Затем применяется алгоритм нахождения устойчивых словосочетаний, который основывается на подсчёте частоты появления слов в сочетаниях и по отдельности. В результате часть слов объединяются в один токен для работы с ними как с одним термином (словосочетания).

Было проведено дополнительное нагрузочное тестирование для анализа производительности процессов индексации и поиска. На основании результатов экспериментов можно сделать заключение, что подход на основе метода TextRank обеспечивает хорошее качество индексирования документной базы и быстрый поиск с ранжированием по ней. При этом для начала работы не требуется предварительно научения и дополнительных наборов доменно-специфичных текстов, что делает качество работы независимым от предметной области и языка, а также от количества хранимых в базе текстов.

ЛИТЕРАТУРА

1. Парамонов А.И. Представление знаний гибридной моделью для систем интеллектуального поиска / Вестник Донецкого национального университета – 2005. – Серия А, №1. – С. 404-409.

2. Stefan Büttcher, Charles L. A. Clarke, Gordon V. Cormack Information Retrieval: Implementing and Evaluating Search Engines / The MIT Press, 2010, 630pp, ISBN 978-0-262-02651-2.

3. Grossi R., Vitter J.S. Compressed suffix arrays and suffix trees with applications to text indexing and string matching, Proceedings on 32nd Annual ACM Symposium on Theory of Computing (STOC 2000), 2000 ACM (pg. 397-406)

4. Eduardo Muñoz, Getting started with NLP: Tokenization, Document-Term Matrix, TF-IDF [Электронный ресурс]. – Aug 3, 2020. Mode of access: https://medium.com/analytics-vidhya/getting-started-with-nlp-tokenization-document-term-matrix-tf-idf-2ea7d01f1942 (last access: 31.01.2022)

5. J. Ramos. Using TF-IDF to Determine Word Relevance in Document Queries. Technical report, Department of Computer Science, Rutgers University, 2003.

6. J. Beall, The weaknesses of full-text searching, Journal of Academic Librarianship, vol. 34, pp. 438-444, 2008.

7. R. Mihalcea and P. Tarau. TextRank: Bringing Order into Texts // Proc. of the 9th Conf. on Empirical Methods in Natural Language Processing. – 2004. – С. 404–411