УДК 004.62, 004.424.62, 004.424.4

СВЯЗЫВАНИЕ ТЕКСТОВЫХ ЗАПИСЕЙ В ЗАДАЧЕ ИНТЕГРАЦИИ ДАННЫХ В УСЛОВИЯХ БОЛЬШИХ ДАННЫХ

Папоян Владимир Владимирович[1], Кореньков Владимир Васильевич[2], Кадочников Иван Сергеевич[3]

[1]Студент;
ГБОУ ВО МО Университет «Дубна»,
Институт системного анализа и управления;
141980, Московская обл., г. Дубна, ул. Университетская, 19;
e-mail: vlpapoyan@jinr.ru.

[2]Директор, доктор технических наук, профессор;
Объединенный институт ядерных исследований,
Лаборатория информационных технологий;
141980, Московская обл., г. Дубна, ул. Жолио-Кюри, 6;
ГБОУ ВО МО Университет «Дубна»,
Институт системного анализа и управления;
141980, Московская обл., г. Дубна, ул. Университетская, 19;
e-mail: korenkov@jinr.ru.

[3]Старший научный сотрудник, инженер-программист;
Российский экономический университет им. Г.В. Плеханова,
Лаборатория «Облачных технологий и аналитики Больших данных»;
117997, г. Москва, Стремянный пер., 36;
Объединенный институт ядерных исследований,
Лаборатория информационных технологий;
141980, Московская обл., г. Дубна, ул. Жолио-Кюри, 6;
e-mail: kadivas@jinr.ru

При интеграции данных из нескольких источников появляется проблема выявления идентичных записей, то есть относящихся к одному и тому же объекту реального окружения. Одно из решений вышеуказанной проблемы осуществляется с помощью вероятностного подхода связывания текстовых записей. В рамках настоящей статьи определено и апробировано, что для эффективной реализации вышеуказанного решения необходимо применить локально-чувствительное хеширование и представить целевой атрибут в векторной модели на этапе блокирования данных. Реализация выявленного подхода была протестирована на двух реестрах компаний Companies House и GLEIF в платформе обработки больших данных Apache Spark.

Ключевые слова: большие данные, связывание текстовых записей, векторное представление слов, локально-чувствительное хеширование, *Apache Spark.*

RECORD LINKAGE IN DATA INTEGRATION PROBLEM UNDER BIG DATA CONDITIONS

Papoyan Vladimir[1], Korenkov Vladimir[2], Kadochnikov Ivan[3]

[1]Student;
Dubna State University,
Institute of the system analysis and management;
141980, Dubna, Moscow reg., Universitetskaya str., 19;
e-mail: vlpapoyan@jinr.ru.

[2]Director, Doctor of Technical Science, professor;
Joint Institute for Nuclear Research,
Laboratory of Information Technologies;
141980, Dubna, Moscow reg., Joliot-Curie, 6;
Dubna State University,

Institute of the system analysis and management;
141980, Dubna, Moscow reg., Universitetskaya str., 19;
e-mail: korenkov@jinr.ru.

[3]Senior researcher, engineer-programmer;
PLEKHANOV Russian University of Economics,
117997, Moscow, Stremyanny lane, 36,
Joint Institute for Nuclear Research,
Laboratory of Information Technologies;
141980, Dubna, Moscow reg., Joliot-Curie, 6;
e-mail: kadivas@jinr.ru

The problem of identifying records refer to the same entity arises appears during the integration data from multiple sources. The application of probabilistic record linkage is one of the key to solve described problem. In this article defined and tried that application of locality-sensitive hashing and vector space model on the blocking stage allow to reach the efficient implementation of described above decision. The implementation is tested in Apache Spark on two registers of companies GLEIF and Companies House.

Keywords: Big Data, record linkage, vector space model, locality-sensitive hashing, Apache Spark.

Введение

Современная банковская деятельность невозможна без использования информационных систем. Банки активно применяют ряд современных информационных технологий для решения различного рода задач.

Научно-исследовательской лабораторией «Облачных технологий и аналитики Больших данных» Российского экономического университета имени Г.В. Плеханова ведется разработка информационного ресурса контроллинга деятельности компаний-нерезидентов для увеличения эффективности заключения финансовых сделок.

На одном из этапов разработки настоящего ресурса требуется объединение множества источников информации о компаниях-нерезидентах. При интеграции данных возникает проблема, когда в итоговом наборе данных содержится несколько записей, описывающих одну и ту же сущность реального мира. В связи с этим возникает необходимость в выявлении похожих записей, относящихся к одному и тому же объекту реального окружения. Решение данной проблемы осуществляется посредством применения вероятностного подхода связывания текстовых записей, описанного в работе [1].

Так как количество компаний порядка нескольких миллионов, то рассмотрение декартового произведения записей о компаниях приводит к большим вычислительным затратам. Вследствие этого в рамках настоящей статьи осуществляется обобщение вероятностного подхода связывания текстовых записей для работы с большим количеством информации.

1. Блокирование в контексте связывания текстовых записей

Связывание текстовых записей является задачей поиска записей в наборе данных, относящихся к одной и той же сущности в различных источниках данных. Например, для человека очевидно, что такие названия компаний как *ASPCIV* (2015) *L.P.* и *ASPCIV* (2015) *L.P* являются идентичными. Однако в процессе сопоставления строк вышеуказанные компании определяются как разные.

Вероятностные алгоритмы связывания текстовых записей были разработаны для оценки вероятности того, что две записи соответствуют друг другу. Для получения вероятности схожести необходимо рассмотрение всех возможных пар записей. Однако, при работе с большими источниками данных рассмотрение всего декартового произведения приводит к большим вычислительным затратам. В статье [1] приведено сокращение пространства сравнений только до тех пар, которые соответствуют определенным критериям. Настоящий подход является блокированием.

Авторы вышеуказанной статьи предлагают группировать данные по имеющимся в источнике атрибутам. Например, в случае поиска идентичных компаний их можно разбить на юрисдикции и уже

внутри каждой юрисдикции осуществлять попарные сопоставления. Однако, при отсутствии в источнике такого рода полей, по которым можно было бы сократить пространство сравнений, наличии в такого рода полей ошибок (например, орфографических), или в случае, когда полученная группа большого размера, происходит возврат к проблеме связанной с вычислительными затратами.

В настоящей статье предлагается конкретизация данного этапа при работе с Большими данными в контексте разработки информационного ресурса контроллинга деятельности компаний-нерезидентов.

2. Выявление подходов для осуществления эффективного блокирования

Основная суть обозначенного подхода состоит в применении локально-чувствительного хеширования.

Locality-Sensitive Hashing (*LSH*) – это вероятностный метод понижения размерности многомерных данных. *LSH* отображает множество точек в высокоразмерном пространстве в множество хеш-таблиц. Идея состоит в том, чтобы подобрать хеш-функции так, чтобы точки данных, которые находятся близко друг к другу, помещались в одни и те же корзины с высокой степенью вероятности, тогда как точки, которые являются удаленными друг от друга, вероятно размещены в разных корзинах [2].

В отличии от традиционных хэшей *LSH* обладает свойством чувствительности к местоположению, благодаря чему способен помещать соседние точки в одну и ту же хэш-таблицу. Таким образом, попарное сопоставление будет осуществляться только для близких точек данных.

Однако для применения *LSH* необходимо представить целевой атрибут в векторном пространстве, так как в рассматриваемом случае он имеет строковый тип. Для того чтобы представить строки в векторной модели, используются *n*-граммы как последовательность из *n* символов и мера *TF-IDF*.

TF-IDF (от англ. *TF – term frequency, IDF – inverse document frequency*) – статистическая мера, используемая для оценки важности слова в контексте документа, являющегося частью коллекции документов. Вес некоторого слова пропорционален частоте употребления этого слова в документе и обратно пропорционален частоте употребления слова во всех документах коллекции [3].

Мера *TF-IDF* часто используется для представления документов коллекции в виде числовых векторов, отражающих важность использования каждого слова из некоторого набора слов (количество слов набора определяет размерность вектора) в каждом документе. Подобная модель называется векторной моделью и дает возможность сравнивать тексты, сопоставляя представляющие их вектора в какой-либо метрике (евклидово расстояние, косинусная мера, расстояние Чебышёва и другие).

Чаще всего термами в *TF-IDF* являются слова. В данном случае использование слов в качестве термов не очень подходит, так как большинство названий компаний содержат только одно или два слова. В связи с этим используются *n*-граммы.

В области компьютерной лингвистики и вероятности *n*-грамма представляет собой непрерывную последовательность из *n* элементов для данного образца текста или речи [4]. С семантической точки зрения *n*-грамма может являться последовательностью звуков, слогов или букв.

В результате расчета меры *TF-IDF* появляется возможность сравнивать строки, сопоставляя представляющие их вектора в какой-либо метрике.

Применение *LSH* к векторной модели целевого атрибута, позволяет произвести блокирование исходного набора данных в меньший набор. Количество пар подлежащих дальнейшему рассмотрению ограничено только теми, между которыми расстояние по какой-либо заранее установленной метрике меньше заданного значения. Например, существует два набора данных *A* и *B*, в каждом из которых 1 000 000 записей, тогда в новый набор данных *C* поступают те пары, у которых евклидово расстояние меньше 10. Те пары записей, которые не соответствуют указанному критерию, автоматически классифицируются как несоответствия и удаляются из рассмотрения.

3. Реализация блокирования посредством применения выявленных подходов

На текущем этапе рассматривается пример реализации обозначенного подхода для поиска близких названий компаний. В рамках выбранного примера в качестве источников данных используются национальный реестр Великобритании *Companies House* [5] и мировой реестр *GLEIF* [6]. Для данных источников осуществляется блокирование данных с целью дальнейшего выявления дублирующихся компаний. Практическая реализация выполняется в платформе параллельной обработки больших данных *Apache Spark* [7].

В рамках реализации используются следующие *Spark* функции:

- *RegexTokenizer*;
- *NGram*;
- *HashingTF*;
- *IDF*.

RegexTokenizer позволяет разбить название компании на последовательность из символов. В результате последовательность символов поступает на вход функции *NGram*. Результатом данной функции является последовательность из триграмм. Далее последовательность триграмм используется для расчета *TF* в функции *HashingTF*. *HashingTF* – это трансформатор, который принимает наборы термов (в данном случае триграммы) и преобразует их в векторы функции фиксированной длины. Полученные вектора используются для построения *IDF*-модели. *IDF* позволяет снизить вес распространенных триграмм и выделает необычные триграммы в строке. *IDF*-модель строится на объединенных данных из реестров *Companies House* и *GLEIF*. В результате вычисляется мера *TF-IDF*. Таким образом, названия компаний в реестрах представлены в виде векторов.

Например, для компании *McDonalds* данный процесс будет выглядеть так, как представлено на рисунке 1.

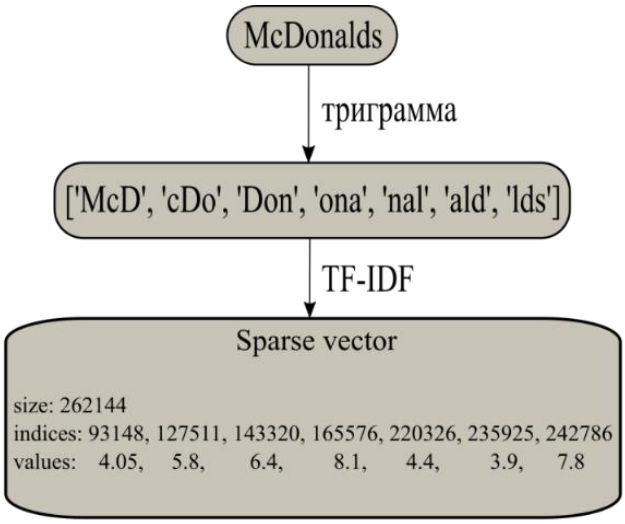

Рис. 1. Векторное представление названия компании McDonalds

Следующий этап – сопоставление векторов. Для осуществления сопоставления применяется подход понижения размерности многомерных данных посредством применения хэш-функций. В *Apache Spark* семейство *LSH* функций реализовано в отдельных классах.

В качестве хеш-функции используется *Bucketed Random Projection* для расстояния Евклида. В семействе *LSH* хеш-функция $h(x)$ описывается как скалярное произведение:

$$h(x) = x \cdot v, \tag{1}$$

где x – это входной вектор, а v – случайный вектор, компоненты которого являются выборкой нормального распределения $N(0, 1)$. Затем осуществляется квантование функции $h(x)$ в набор хеш-блоков для того, чтобы близкие элементы попали в одну и ту же корзину, то есть:

$$h(x) = \frac{x \cdot v}{r}, \quad (2)$$

где r – длина корзины, которая регулирует средний размер хеш-блоков и, следовательно, количество блоков.

Входными данными являются разреженные вектора, каждый из которых представляет точку в евклидовом пространстве. Далее применяется функция *Approximate similarity join* для определения сходства. Вышеуказанная функция возвращает набор данных, в котором содержатся такие пары строк, расстояние которых меньше установленного порога.

На текущем этапе рассматривается пример сопоставления векторов из реестров *Companies House* и *GLEIF*. Для построения модели, использовалась длина корзины со значением 1 и три хэш-таблицы. Порог для функции *Approximate similarity join* был установлен в значении 20. Таким образом, были найдены пары компаний, у которых расстояние по Евклиду было меньше чем 20. Гистограмма распределения расстояния представлена на рисунке 2.

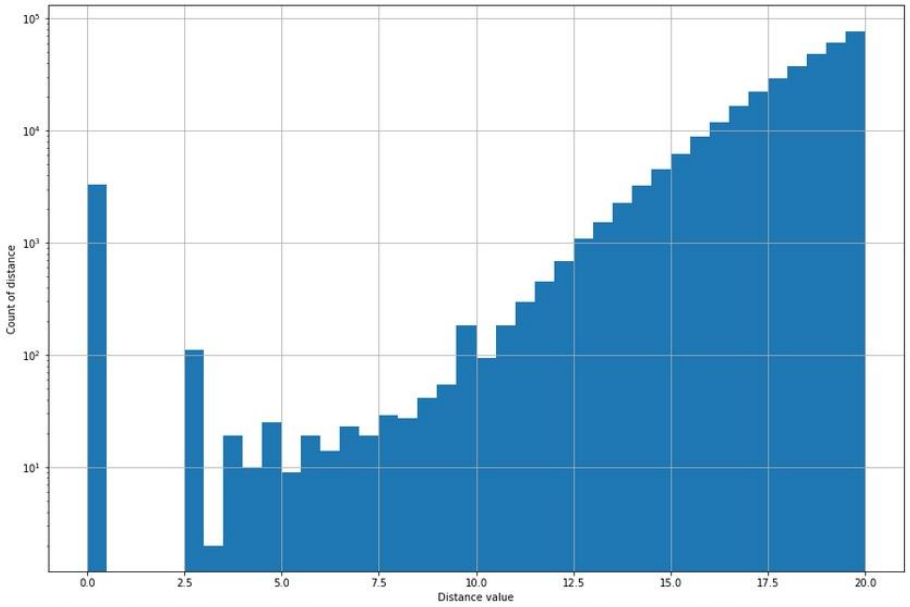

Рис. 2. Распределение строкового расстояния для кандидатов в соответствие

Данные рисунка 2 позволяют определить, что, начиная со значения 10, количество компаний растет, то есть все большее число компаний имеет дальнее расстояние на каждом промежутке.

Заключение

Таким образом, из большого множества компаний были получены такие пары, у которых евклидово расстояние для названий компаний меньше 20. Количество записей в новом наборе данных составило примерно $3 \cdot 10^5$. Количество всех возможных пар компаний порядка $4 \cdot 10^{12}$, так как размерность реестра *Companies House* составляла $4 \cdot 10^6$, а реестр *GLEIF* содержит примерно 10^6 записей. Следовательно, пространство сравнений было сокращено в 10^7 раз за счет хеширования.

В результате последовательного выполнения последующих этапов вероятностного подхода связывания текстовых записей на полученном наборе данных осуществляется выявления идентичных компаний.

Описанный в настоящей статье подход позволяет применять вероятностный метод связывания текстовых записей при работе с Большими данными в том случае, если возможность группировки по одной из причин, описанных в п. 1, отсутствует.

Таким образом, в рамках разработки информационного ресурса контроллинга деятельности компаний-нерезидентов на этапе интеграции данных о компаниях осуществлено выявление идентичных по своему смыслу записей.

Подход по блокированию данных, описанный в настоящей статье, применим не только к информации из реестров компаний, но и к аналогичным, схожим по своей структуре источникам данных, отличающимся по содержанию в зависимости от рассматриваемой проблематики.

Список литературы

1. Sayers A., Ben-Shlomo Y., Blom A. W., Steele F. Probabilistic record linkage. International Journal of Epidemiology, 2016. – Vol. 6. P. 954-964.
2. Jure Leskovec, Anand Rajaraman, Jeff Ullman. Mining of Massive Datasets.: Cambridge University Press, 2014. – Глава. 3.4.
3. ZHANG Yun-tao, GONG Ling, Wang Yong-cheng. Journal of Zhejiang University SCIENCE, 2005. – Vol. 45. – Issue. 1. P. 49-55. ISSN: 1009-3095.
4. William B. Canvar, John M. Trenkle. Proceedings of SDAIR-94, 3rd Annual Symposium on Document Analysis and Information Retrieval, Las Vegas, US, 1994. P. 161-175.
5. Companies House [Электронный ресурс] // Free Company Data Product – Режим доступа: http://download.companieshouse.gov.uk/en_output.html, свободный (дата обращения 01.07.2019).
6. LEI Data [Электронный ресурс] // Download the Concatenated Files – Режим доступа: https://www.gleif.org/en/lei-data/gleif-concatenated-file/download-the-concatenated-file/, свободный (дата обращения 01.07.2019).
7. Apache Spark [Электронный ресурс] // Unified analytics engine for large-scale data processing. – Режим доступа: https://spark.apache.org/, свободный (дата обращения 01.07.2019).