УДК 004.6

Методы и средства интеграции и обогащения библиотечных данных

К.А. Косолапов

Московский государственный университет им М.В. Ломоносова

В современное время мало какая библиотека хранит данные о книгах в бумажном виде, повсеместно можно видеть использование электронных баз с библиотечными данными. Основной целью данной работы является создание программной системы, которая бы в автоматическом режиме собирала библиографические записи из различных библиотек, находила дубликаты записей и проводила слияние их в общем хранилище.

Выполняя свои функции, библиотеки создают библиографические записи на экземпляры, хранящиеся в их фондах. Общее количество записей о книгах в крупных библиотеках может достигать 10 миллионов. Как следствие, в библиотечных системах хранятся записи на одни и те же произведения. При этом в каждой из библиотек формат хранения и требования к полноте заполнения записей могут отличаться.

Для выявления дубликатов библиографических записей, полученных из разных источников и заполненных с использованием различных стандартов, требуется создание эффективного алгоритма. При получении нескольких десятков миллионов записей из нескольких библиотек задача сравнения всех записей со всеми для выявления дублетных записей на одни и те же книги становится неприемлемо затратной – $O(n^2)$. Возникает необходимость создания алгоритма, позволяющего существенным образом сузить множество библиографических записей – потенциальных дубликатов. Поскольку зачастую дополнительная информация, указанная в записи в виде ссылок, может быть недоступна, например, получить полный текст произведения по ссылке из описания невозможно из-за ограничений, накладываемых авторским правом, то процесс выявления дублетных записей может ориентироваться только на информацию, содержащейся в самой записи. В данной работе представлен алгоритм поиска неточных дубликатов для коротких текстовых полей.

В отличие от строк, поиск и сравнение среди большого количества натуральных чисел можно организовать более эффективно. Достаточно разместить их в отсортированном массиве. Тогда, используя алгоритм двоичного поиска, можно найти необходимое число за $O(\log n)$ операций сравнения. Для перевода строковых значений в числовые используются функции хэширования. Однако, большинство алгоритмов хэширования не подходят для нашей задачи, так как изначально они создавались для равномерного отображения пространства строк на пространство чисел. Это означает, что

при незначительном изменении аргумента функции хеширования – строки, значение функции – число может измениться разительно. Описываемый в работе алгоритм основывается на схеме хэширования SimHash [1], которая позволяет получать достаточно близкие в смысле расстояния Хэмминга значения хэшей.

Поскольку количество записей велико, то сгенерировав хэши для всех записей, появляется проблема эффективного поиска среди этого набора, поскольку в худшем случае для поиска близких хэшей необходимо произвести 2^{32} операций сравнения. Для снижения количества сравнений был использован метод разбиения хэша на равные части, описанный в работе Bingfeng Pi, Shunkai Fu, Weilei Wang и Song Han [2]. Также предложена оптимизация работы алгоритма, основывающаяся на допущении о локализации ошибки среди дубликатов.

Для выявленных дубликатов производится операция слияния. Слияние осуществляется по свойствам формата MODS, к которому приводятся все различные форматы данных из библиотек. Для библиографической записи в формате MODS существуют простые и составные свойства. Для составных свойств происходит объединение наборов данных из различных записей. В случае если простые свойства не совпадают, используется значение простого свойства из более старой записи.

Полученный алгоритм является оптимизацией известных алгоритмов поиска нечетких дубликатов для случаи большого количества исходных данных. Созданная на его основе программная система обладает высокой производительностью. Последующие исследования могут быть направлены на создание масштабируемого алгоритма, способного разделять библиографические записи на несколько потоков и параллельно их обрабатывать.

Литература

1. Charikar M.S. Similarity estimation techniques from rounding algorithms / M. S. Charikar // Proc. thiry-fourth Annu. ACM Symp. Theory Comput. - STOC ’02 – 2002. – 380–388c.
2. Pi B. SimHash-based Effective and Efficient Detecting of Near-Duplicate Short Messages / B. Pi, S. Fu, W. Wang, S. Han // Science (80-.). – 2009. – T. 7 – 20–25c.