MINISTRY OF EDUCATION AND SCIENCE OF UKRAINE
SUMY STATE UNIVERSITY
UKRAINIAN FEDERATION OF INFORMATICS

PROCEEDINGS

OF THE IV INTERNATIONAL SCIENTIFIC CONFERENCE

ADVANCED INFORMATION SYSTEMS AND TECHNOLOGIES

AIST-2016

May 25 –27, 2016
Sumy, Ukraine

Review of Methods of Normalization of a Text for Handling

T. V. Plavin
National Technical University of Ukraine "Kyiv Polytechnic Institute", Ukraine, taras.plavin@gmail.com

Abstract. The article deals with the issues of normalization of a text. It outlines steps needed to be done for quality normalization.

Keywords. Text Comparison, Text Search, Normalization, Natural Language, Plagiat, Text Segmentation.

ВВЕДЕНИЕ

Сейчас существует много систем которые работают с текстами. Очень важным этапом при работе с текстами является его нормализация, ведь она позволяет улучшить качество обработки и в конечном итоге выдать более точный результат. Сам процесс нормализации можно разбить на много частей и в каждом конкретном случае могут быть свои особенности. В данной работе будут рассмотрены основные этапы нормализации для поиска и сравнения текстов.

ОСНОВНАЯ ЧАСТЬ

Перед началом сравнения текстов над ними нужно провести определенную работу. Сперва имеется документ в определенном формате, который нужно разобрать. Как правило каждый формат имеет свою специфику, но при разборе любого формата нужно убрать оглавление, таблицы, список литературы и титульные страницы (они содержат короткие, как правило служебные, строки, которые только увеличивают объем текста и мешают сравнению), убрать картинки и пометить интернет адреса. Для HTML страниц можно исключить колонтитулы и боковые колонки.

Затем нужно заменить все высокохудожественные юникодовские символы на простые — кавычки ёлочкой, кавычки в виде перевернутых запятых, длинные и полудлинные тире, апострофы, троеточия. Заменить два апострофа подряд на нормальные кавычки, а два тире — на одно. Все последовательности пробельных символов заменить на один обычный пробел. И напоследок убрать все управляющие символы, кроме обычного перевода строки [3].

Затем нужно найти кириллические и латинские символы похожие по написанию, например английские буквы "B", "p", "c", "e" и тд. в русских словах и буквы русского алфавита в английских словах. Алгоритм такого поиска довольно простой, хотя не стопроцентный. Зато он позволяет не использовать базу данных со словами определенного языка и учитывает то, что текст может содержать слова с разных языков. Сначала должна быть создана база данных с соответствиями похожих кириллических и латинских символов. Затем выполнить следующие шаги:

1) Получается исходное слово
2) Проверяется, есть ли в нем одновременно и кириллические и латинские символы, если нет – можно переходить к следующему слову.
3) Заменяются все латинские символы на кириллические, используя базу данных с соответствиями. Происходит переход к шагу два. Если проверка не прошла, заменяются все кириллические символы на латинские и заново происходит переход к шагу 2.

Порядок замены латинских символов на кириллические или наоборот лучше менять, в зависимости от того какие тексты чаще обрабатываются. Алгоритм не сработает на

словах, которые полностью содержат похожие кириллические и латинские символы, если используется не подходящий порядок замены. Например имеется слово «Bop», в котором символ «p» - это латинская буква. Если на третьем шаге будут заменены все кириллические символы на латинские, то получится слово с полностью латинскими буквами и проверка на шаге 2 даст положительный результат. Обрабатывать такие случаи можно по разному, немного модифицировав алгоритм.

Еще одним механизмом, который улучшает сравнивание текстов, является стемминг — это процесс нахождения основы слова для заданного исходного слова. Основа слова необязательно совпадает с морфологическим корнем слова. Задача нахождения основы слова представляет собой давнюю проблему в области компьютерных наук. Стемминг в основном применяется в поисковых системах для расширения поискового запроса пользователя и является частью процесса нормализации текста.

Использование стемминга может улучшать поиск и сравнение текстов. Пример поиска, в котором стемминг дает результат: пользователь ищет по слову «fish», но также получает результаты в которых содержится слово “fishing” и наоборот. В сравнении текстов плагиаторы часто меняют слова местами, а соответственно меняются падежи и окончания слов. Если не использовать стемминг, сравнение не даст положительных результатов, ведь технически будут сравниваться два разных слова. Есть много алгоритмов стемминга, в том числе для украинского и русского языков [1].

Еще одним, не мало важным в процессе сравнения текстов, этапом является разбиение на предложения, который в зарубежной литературе носит название “text segmentation”, ведь конечное сравнение происходит именно по предложениям. При разбиении на предложения, нужно учитывать, что в них могут быть сокращенные слова, веб-адреса, цитаты, дробные числа и тд., то есть просто разбиение на предложения по точкам, знакам вопроса и знакам восклицания не даст нужный результат. Для решения этой задачи можно использовать регулярные выражения и базу данных с сокращениями, которая будет содержать типовые сокращения и части веб-адресов, например «.com», «см.», «тыс.» и тд. При правильном проектировании регулярных выражений можно покрыть почти все случаи, при условии, что текст оформлен корректно [2].

ВЫВОДЫ

В данной работе были рассмотрены способы и алгоритмы нормализации, которые в общей сумме значительно улучшают качество сравнения и поиска текстов.

При использовании этих процессов обработки текста можно создавать различные системы для работы с текстом - онлайн сервисы, десктоп или мобильные приложения, например сервисы для разбиения текста на предложения, параграфы или другие структурные единицы, поиска плагиата, сравнения текстов, поиска в текстах. Также они могут выводить статистику обработки, например количество совпадений, количество замен, веб адреса, семантический анализ и тд.

REFERENCES

[1] Ilia Smirnov Overview of stemming algorithms // Mechanical Translation. — 2008.

[2] Freddy Y. Y. Choi (2000). "Advances in domain independent linear text segmentation". Proceedings of the 1st Meeting of the North American Chapter of the Association for

[3] Daniel Jurafsky, James H. Martin Speech and Language Processing. An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition. Second Edition . Pearson Education International . -2009 – 1024 pp.