Интеллектуальное сравнение текстов для поиска плагиата

Тямгин И.А. Белозуб В.А. Аметов О.З.

28 декабря 2014 г.

Аннотация

Целью данной работы было исследование алгоритмов сравнения текстов для поиска плагиата в русскоязычном тексте и реализация нескольких из этих алгоритмов. Некоторые из рассмотренных алгоритмов были реализованы с небольшими изменениями и упрощениями.

Введение

Большие объемы обрабатываемых данных делают задачу поиска похожих текстов алгоритмически сложной. При этом алгоритмы, успешно работающие для конкретной постановки задачи, бывают неприменимы или дают плохие результаты для других задач. Так, например, в юрислингвистике для определения авторства текста изучают его стилистические особенности, категоричность высказываний, использование оценочной лексики. Важно четко определить понятия плагиата и похожих текстов.

1 Понятие плагиата

Плагиат — умышленное присвоение авторства чужого произведения искусства или достижения науки, технических решений или изобретений. Плагиат может быть нарушением авторско-правового законодательства и патентного законодательства и в качестве таковых может повлечь за собой юридическую ответственность. С другой стороны, плагиат возможен и в областях, на которые не распространяется действие каких-либо

видов интеллектуальной собственности, например, в математике и других фундаментальных научных дисциплинах. [1]

В данной работе будем придерживаться именно этого определения, поскольку оно в той или иной форме приводится в большинстве статей, описывающих системы и алгоритмы определения плагиата.

Для разработки алгоритма важно определить два аспекта:

- Критерий определения похожести текстов (форма или содержание)
- Определение оценки степени похожести и ее порогового значения, до которого тексты не считаются дубликатами.

Согласно классификации, приведенной в [3], случаи плагиата можно разделить на две основные группы: точное копирование и копирование с модификацией. Во втором случае текст может быть переформулирован или переведен на другой язык. Кроме того, может быть скопирован как целый текст, так и его часть. Алгоритмы, успешно обнаруживающие плагиат одного типа, могут давать очень слабые результаты для плагиата других типов.

2 Постановка задачи

Целью данной работы является исследование и разработка методов поиска плагиата в русскоязычных текстах. Для достижения данной цели были поставлены следующие задачи:

1. Исследовать существующие методы поиска плагиата.
2. Реализовать систему ставнения текстов, подтверждающую работоспособность данных методов.

3 Обзор существующих методов

Существует ряд методов для нахождения плагиата в текстах. Наиболее часто процесс поиска осуществляется по следующему алгоритму:

1. Берут два текста, подозрительные на плагиат.
2. Для этих двух документов анализируется степень сходства с проверяемым, если она достаточно высока – предполагается случай плагиата.

3. Пост-обработка. Полученные результаты проверяются, чаще всего вручную, для исключения ложных обнаружений (например, случаев, когда за плагиат принимается оформленная по всем правилам цитат).

3.1 Алгоритм шинглов

Одним из наиболее часто используемых является метод «шинглов», предложенный в 1997 году. Он основан на представлении документа в виде всевозможных последовательностей фиксированной длины k, состоящих из соседних слов. Такие последовательности назвали «шинглами» (от англ. shingles). Два документа считаются похожими, если множества их шинглов существенно пересекаются. [2]

Данный подход можно разбить на такие основные этапы:

- канонизация текста
- разбиение на шинглы
- вычисление хэшей шинглов
- сравнение, определение результата

Канонизация текста. Канонизация текста приводит оригинальный текст к единой нормальной форме. Текст очищается от предлогов, союзов, знаков препинания, HTML тегов, и прочего ненужного «мусора», который не должен участвовать в сравнении. В большинстве случаев также предлагается удалять из текста прилагательные, так как они не несут смысловой нагрузки. Также на этапе канонизации текста можно приводить существительные к именительному падежу, единственному числу, либо оставлять от них только корни.

Разбиение на шинглы. *Шинглы (англ — чешуйки)* — выделенные из статьи подпоследовательности слов. Необходимо из сравниваемых текстов выделить подпоследовательности слов, идущих друг за другом по k штук (длина шингла). Выборка происходит внахлест, а не встык. Таким образом, разбивая текст на подпоследовательности, мы получим набор шинглов в количестве равному количеству слов минус длина шингла плюс один.

Вычисление хэшей шинглов. Принцип алгоритма шинглов заключается в сравнении случайной выборки контрольных сумм шинглов (подпоследовательностей) двух текстов между собой. Проблема алгоритма заключается в количестве сравнений, ведь это напрямую отражается на

производительности. Увеличение количества шинглов для сравнения характеризуется экспоненциальным ростом операций, что критически отразится на производительности.

3.2 I-Match

Рассмотрим другой сигнатурный подход, основанный уже не на синтаксических, а на лексических принципах. Для этого понадобиться понятия TF и IDF [4].

TF (term frequency — частота слова) — отношение числа вхождения некоторого слова к общему количеству слов документа. Таким образом, оценивается важность слова в пределах отдельного документа.

$$\mathrm{tf}(t, d) = \frac{n_i}{\sum_k n_k} \tag{3.1}$$

где n_i есть число вхождений слова в документ, а в знаменателе — общее число слов в данном документе.

IDF (inverse document frequency — обратная частота документа) — инверсия частоты, с которой некоторое слово встречается в документах коллекции. Учёт IDF уменьшает вес широкоупотребительных слов. Для каждого уникального слова в пределах конкретной коллекции документов существует только одно значение IDF

$$\mathrm{idf}(t, D) = \log \frac{|D|}{|(d_i \supset t_i)|} \tag{3.2}$$

где

- |D| — количество документов
- $|(d_i \supset t_i)|$ - — количество документов, в которых встречается t_i.

Выбор основания логарифма в формуле не имеет значения, поскольку изменение основания приводит к изменению веса каждого слова на постоянный множитель, что не влияет на соотношение весов.

Основная идея этого подхода, описанного в [3], состоит в вычислении дактилограммы I-Match для представления содержания документов. С этой целью сначала для исходной коллекции документов строится словарь L, который включает слова со средними значениями IDF. Слова с большими и маленькими значениями IDF отбрасываются. Затем для каждого документа формируется множество U различных слов, входящих в него, и определяется пересечение U и словаря L. Если размер этого пересечения больше некоторого минимального порога (определяемого

экспериментально), то список слов, входящих в пересечение упорядочивается, и для него вычисляется I-Match сигнатура (hash-функция).

Два документа считаются похожими, если у них совпадают I-Match сигнатуры. Алгоритм имеет высокую вычислительную эффективность, превосходящую показатели алгоритма шинглов. Другим преимуществом алгоритма (также, по сравнению с алгоритмом шинглов) является его высокая эффективность при сравнении небольших по размеру документов. Основной недостаток — неустойчивость к небольшим изменениям содержания документа.

3.3 Слова с оптимальным весом

Алгоритм реализует метод «оптимальной поисковой частоты», предложенный М. Масловым, и использующийся для поиска похожих документов в широком спектре приложений, от веб-поисковика до кластеризации новостей. Суть его заключается в следующем. Вместо классической метрики $TF * IDF$ предлагается ее модифицированный вариант.

Вводится эвристическое понятие «оптимальной частоты» для слова равное

$$\ln \frac{10}{1000000} = 11.5 \tag{3.3}$$

Т.е. «оптимальным» считается вхождение слова в 10 документов из 1000000. Если реальное значение IDF меньше «оптимального», то оно немного (по закону параболы) повышается до

$$IDF_{opt} = \sqrt{\frac{IDF}{11.5}} \tag{3.4}$$

а если больше, то существенно (как гипербола) снижается до

$$IDF_{opt} = \frac{11.5}{IDF} \tag{3.5}$$

Таким образом по всей коллекции строится словарь, ставящий каждому слову в соответствие число документов, в которых оно встречается хотя бы один раз (df). Далее строится частотный словарь документа и для каждого слова вычисляется его «вес» w_t по формуле:

$$w_t = TF * IDF_{opt} \tag{3.6}$$

$$TF = 0.5 + 0.5 * \frac{tf}{tf_{max}} \tag{3.7}$$

$$IDF = -\log \frac{tf}{N} \tag{3.8}$$

$$IDF_{opt} = \sqrt{IDF}11.5 \tag{3.9}$$

если IDF меньше чем 11.5, иначе

$$IDF_{opt} = \frac{11.5}{IDF} \tag{3.10}$$

Затем выбираются и сцепляются в алфавитном порядке в строку слов с наибольшими значениями w_t.

В качестве сигнатуры документа вычисляется контрольная сумма CRC32 полученной строки.

3.4 Наиболее часто встречающиеся слова

В качестве сигнатуры документа используется хэш-код строки, полученной сцеплением k наиболее часто встречающихся в документа слов. Данный метод можно считать одним из случаев использования анализа ключевых слов текстов. Несмотря на свою простоту, при использовании нормализации и стоп-слов метод дает неплохие результаты.

4 Практическая часть

В качестве языка был выбран Python 2.7, являющийся кроссплатформенным языком, подходящим для работы со строками и регулярными выражениями, а также с протоколом http.

Из перечисленных выше методов выбраны два: алгоритм шинглов и I-Match.

Программа представляет собой веб-страницу, содержащую 2 текстовых поля ввода для сравниваемых документов, combobox для выбора длины шингла. Результат работы - вывод документов с выделенными фрагментами текста, которые присутствуют в обоих документах, которые показал алгоритм шинглов. Аналогично для I-Match.

Реализацию можно взять из репозитория https://github.com/tyamgin/IPI.git

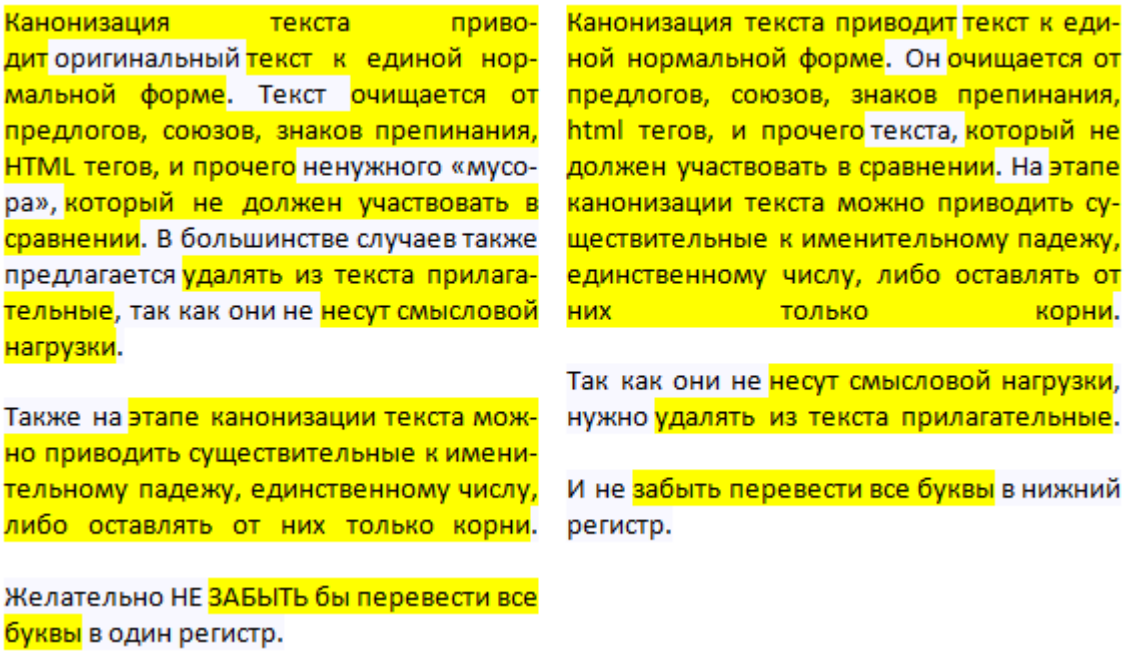

Рис 1. Результат работы алгоритма шинглов

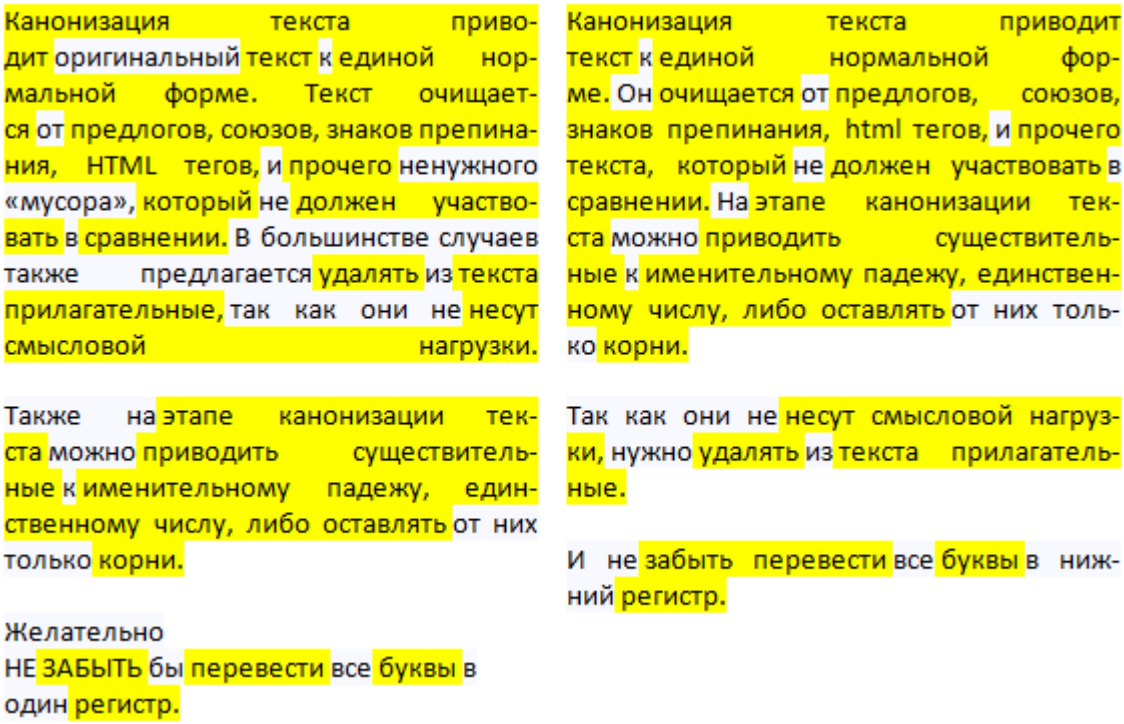

Рис 2. Результат работы алгоритма I-Match

5 Заключение

В рамках данной работы были получены следующие результаты:

1. Исследованы существующие методы поиска плагиата.
2. Построена система сравнения двух текстов, реализующая некоторые из рассмотренных подходов.

Список использованных источников

1. Определение плагиата. https://ru.wikipedia.org/wiki/Плагиат
2. Алгоритм шинглов. https://ru.wikipedia.org/wiki/Алгоритм_шинглов
3. Зеленков Ю.Г., Сегалович И.В. Сравнительный анализ методов определения нечетких дубликатов для WEB-документов // Труды 9-ой Всероссийской научной конференции «Электронные библиотеки: перспективные методы и технологии, электронные коллекции» RCDL'2007: Сб. работ участников конкурса, том 1. Переславль-Залесский, Россия: «Университет города Переславля», 2007. С. 166-174
4. TF-IDF. https://ru.wikipedia.org/wiki/TF-IDF