Алгоритм контекстно-зависимого аннотирования

© Александр Салтыков, Сергей Куротченко, Роман Дорохин

ROOKEE

{alexander.saltikov, sergey.kurotchenko,

roman.dorohin}@re-actor.ru

Аннотация

В этой статье авторы предлагают алгоритм контекстно-зависимого аннотирования документов, в основе которого заложено разбиение контента на пассажи, их кластеризация и ранжирование кластеров по взвешенной оценке параметров значимости пассажей. Алгоритм учитывает вхождения слов, очень близких по смыслу к словам запроса, а также транслитерации слов запроса.

1. Введение

Аннотирование веб-документов является важным критерием качества любой поисковой системы. Аннотирование подразумевает составление краткой аннотации (сниппета) ограниченной длины (обычно это текст длиной 150 - 400 символов). Такая аннотация должна представить пользователю поисковой системы наиболее полную и ценную информацию о странице в соответствии с введенным запросом.

2. Описание алгоритма аннотирования

Особенность предлагаемого алгоритма заключается в том, что при составлении аннотации существенную роль играют не только сами слова запроса, но и слова, очень близкие по смыслу к словам запроса (назовем их слова-переходы), а также для некоторых слов запроса и транслитерации, то есть написание этих слов латиницей. Для составления базы таких слов-переходов использовался специальный алгоритм морфологического анализа и поиска однокоренных слов и слов-синонимов.

Для составления хороших аннотаций предварительно производится выделение из структурных элементов DOM-модели HTML-документа релевантных и значимых блоков текстового контента. Для составления аннотации не учитываются служебные блоки типа меню и др. В результате остается серия текстовых блоков без дополнительной разметки, содержимое которых наиболее полно отражает основное содержание анализируемого документа, по которому и строится аннотация документа. Блок-схема алгоритма выделения структурных элементов представлена на рисунке 1. Алгоритм выделения значимых блоков следующий.

Шаг 1. Обработчику (программе, реализующей исполнение алгоритма) указывается url адрес анализируемой страницы; по указанному адресу производится загрузка HTML-документа.

Шаг 2. Преобразование HTML в XML. Исходный HTML-документ средствами библиотек расширения выбранного языка программирования преобразуется из HTML формата в формат XML, более удобный для последующей обработки.

Шаг 3. Предварительная редукция структуры XML-документа. Производится предварительное преобразование структуры документа, подготавливающее её к последующему сравнительному анализу и извлечению релевантного/значимого содержимого структурных элементов модели. Этап включает: удаление всех не значимых текстовых и блочных элементов, преобразование иерархической структуры документа заменой каждого из её узлов контент-узлом.

Шаг 4. Редукция структуры документа по индикаторам. На этапе редукции структуры по индикаторам реализуется основной процесс отбора релевантного/значимого контента, разбиваемый на три параллельных процесса: редукция структуры документа последовательным понижением числа уровней вложенности; выборочное слияние/отбрасывание листьев дерева документа.

Шаг 5. Извлечение текста из документа. Результатом успешного и корректного выполнения предыдущих шагов алгоритма является документ, состоящий из одного корневого контент-узла, содержащего в себе релевантный/значимый текстовый контент исходного HTML-документа. На пятом шаге работы алгоритма содержимое корневого контент-узла полностью редуцированной модели документа выводится в текстовый файл.


Начало
1
Исходные данные
анализируемый документ;
url анализируемого документа
2
Преобразование HTML в XML
3
Предварительная редукция структуры XML документа; определение nLevels, nNodes
перечень тэгов и типов контента HTML-страниц, подлежащих удалению
4
Редукция структуры документа по индикаторам
база лемм коммерческих слов-маркеров;
база лемм навигационных слов-маркеров;
документы-окружение текущего документа в структуре сайта
5
Извлечение текста из документа
Конец

Рисунок 1. Алгоритм выделения значимых и релевантных структурных элементов для последующего формирования аннотации.

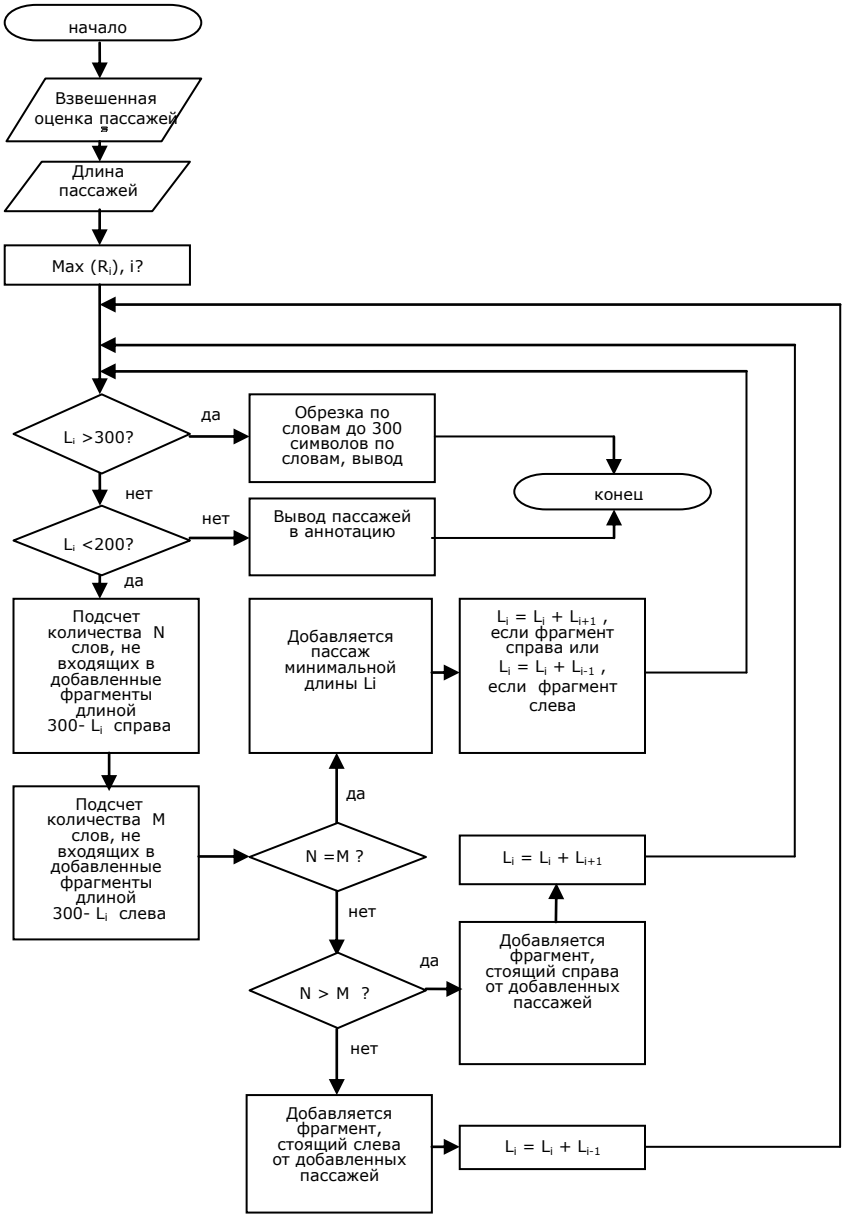

Рисунок 2. Блок-схема алгоритма обрезки аннотации.

После выделения значимых и релевантных структурных элементов документа производится аннотирование документа.

Алгоритм составления аннотации включает в себя следующие шаги:

1. Получение контента значимых и релевантных элементов документа.
2. Разбиение контента на пассажи. Под пассажем здесь понимается фрагмент текста, содержащий законченную мысль. Разбиение текста на пассажи происходит по знакам препинания, обозначающим конец предложения, а также по так называемым ”разделяющим” html-тегам, такими как <p>, </br>, ,<li>, <td> и некоторым другим.
3. Формирование кластеров из N пассажей.
4. Расчет параметров ранжирования пассажей и взвешенной оценки ранга каждого кластера.
5. Ранжирование кластеров по взвешенной оценке и определение кластера пассажей с наибольшим рангом.
6. Запуск алгоритма обрезки аннотации до $n \leq k$ символов (в данном случае $k = 300$) и формирование аннотации. Блок-схема алгоритма обрезки аннотации представлена на рисунке 2.

Алгоритм выбора пассажей для аннотации реализован следующим образом:

1. Все пассажи исходной страницы группируются в кластеры по N пассажей в кластере с шагом в 1 пассаж. В настоящем алгоритме значение N = 4. То есть в первый кластер входят по порядку следования в тексте 1, 2, 3 и 4 предложения, во второй кластер – 2,3,4,5 предложения и т.д.
2. Для каждого пассажа с номером i рассчитываются взвешенный параметр ранжирования R_i по формуле:

$$R_i = \frac{4 \cdot \frac{Ukwp_i}{Uwq} + 1{,}8 \cdot \frac{Tr_i}{Uwq} + 0{,}1 \cdot \mathrm{Ds_i}}{4 + 1{,}8 + 0{,}1} \qquad (1)$$

где $Ukwp_i$ – unique key word passage – количество уникальных слов запроса в i-ом пассаже, стоящие в той же словоформе, что и слова запроса (количество точных вхождений слов запроса в слова каждого предложения).

Uwq – unique word query – количество уникальных слов запроса, исключая стоп-слова (предлоги, союзы и т.д.)

Tr_i – количество уникальных слов-транслитераций и слов-переходов, найденных в каждом пассаже для всех слов запроса.

Ds_i –Наличие хотя бы одного знака “тире” или ”дефис” в пассаже (булевский параметр: 1 – в пассаже есть хотя бы 1 “тире” или ”дефис”, 0 – таких знаков не обнаружено).

3. Для каждого кластера, состоящего из N пассажей, рассчитывается сумма S_i:

$$S_i = \sum_{i=1}^{N} R_i \qquad (2)$$

4. Выбирается кластер пассажей с максимальным значением S_i

5. Запускается алгоритм обрезки выбранного кластера пассажей до 300 символов.

6. Формирование аннотации: Вывод пассажей выбранного кластера в том порядке, в котором эти пассажи идут в тексте. Если соседние пассажи были выделены по разделяющим тегам, в конце рассматриваемого пассажа нет знака препинания ”точка” и если следующий пассаж начинается с большой буквы, то происходит добавление точки в конце рассматриваемого пассажа.

Алгоритм обрезки аннотации работает следующим образом:

1. На вход алгоритма поступает кластер пассажей с максимальным значением S_i, содержащая выбранные для аннотации пассажи, для каждого из которых вычислены: взвешенная оценка по параметрам R_i , Длина каждого пассажа в символах L_i , требуемая предельная длина аннотации k =300 символов.

2. Из пришедших на вход алгоритма пассажей выбирается один пассаж, имеющий максимальную взвешенную оценку R_i.

3. Если количество символов уже добавленных в аннотацию пассажей превышает лимит в 300 символов, то пассаж обрезается по целому слову и после него ставится многоточие таким образом, чтобы количество символов обрезанного пассажа не превышало 300. Искомая аннотация построена.

4. Если количество символов уже добавленных в аннотацию пассажей меньше 300, но больше 200 символов, то в аннотацию этот пассаж выводится целиком. Искомая аннотация построена.

5. Если количество символов этого пассажа меньше 200, то анализируются 2 пассажа слева и справа от уже добавленного в аннотацию пассажа. Для обоих пассажей находится количество уникальных слов запроса, слов-транслитераций и слов-переходов, не

вошедших в уже выбранный пассаж (пассажи) с длиной = 300 – длина уже добавленных пассажей.

6. Для добавления в аннотацию выбирается тот фрагмент текста, который имеет большее число слов, не вошедших в уже добавленный ранее фрагмент аннотации таким образом, чтобы общая длина аннотации не превышала 300 символов. Искомая аннотация построена.

7. Если количество уникальных слов запроса, слов-транслитераций и слов-переходов, не вошедших в уже добавленные в аннотацию для пассажей во фрагментах слева и справа равно (в том числе, и равно нулю), то выбирается пассаж с минимальной длиной, при проверке условия: длина всей аннотации не больше 300 символов. Искомая аннотация построена. Иначе, если длина всей аннотации больше 300, оставляются только целые и уже добавленные в аннотацию пассажи. Искомая аннотация построена.

8. Иначе переход к пункту 4.

На рисунке 2 представлена блок-схема алгоритма обрезки аннотации

3. Результаты и заключение

На рисунках приведено сопоставление результатов участников дорожки контекстно-зависимого аннотирования РОМИП 2010.

Рисунок 3. Сравнительные результаты качества контекстно-зависимого аннотирования.

Как видно из рисунка 3, оба алгоритма выполнения контекстно-зависимого аннотирования получили высокие оценки со стороны

независимых экспертов – асессоров – и эти оценки практически идентичны: разница не превышает 2,5%, что либо сопоставимо, либо меньше достижимой на практике погрешности формирования оценок.

Высокая близость оценок лидеров с двумя полностью независимыми технологиями позволяет сделать вывод о близком к наивысшему в рамках данной системы рейтингов качестве аннотирования, что, безусловно, является достойным результатом.

4. Выводы

К достоинствам предложенного алгоритма можно отнести хорошую читаемость аннотации – аннотация часто содержит целые законченные предложения, наиболее полно формирующие представление о странице по введенному запросу. Также достоинством алгоритма является его простота и высокая скорость работы.

Как показали исследования, полученные по данному алгоритму аннотации, в 50% случаев содержат фрагмент из аннотации, которую формирует поисковая система Яндекс по тому же запросу.

Алгоритм может быть использован для составления аннотаций различных коллекций документов.

Литература

[1] ROMIP 2009. http://romip.ru/ru/2009/tracks/annotation.html

[2] В. Васильев. Выделение фрагментов в текстах при классификации, 2008. http://www.dialog-21.ru/dialog2009/materials/html/10.htm

[3] Christopher D. Manning, Prabhakar Raghavan, Hinrich Schütze, 2008. Introduction to Information Retrieval

ROOKEE annotation algorithm

© Alexander Saltikov, Sergey Kurotchenko, Roman Dorohin

In this article, the authors propose an algorithm for context-dependent annotation of documents, built on a partition of the content on passages of their clustering and ranking of clusters by a weighted estimation of parameters of relevance passages. The algorithm takes into account the occurrence of words that are very close in meaning to the words of the query, as well as transliteration and word-conversion.