Алгоритмы контекстно-зависимого аннотирования Яндекса на РОМИП-2008

© Алексей Бродский, Руслан Ковалев, Михаил Лебедев,

Дмитрий Лещинер, Павел Сушин

Яндекс
{abrodskiy, velavokr, mlebedev, leshch,
psushin}@yandex-team.ru
Илья Мучник
Школа анализа данных
muchnikilya@yahoo.com

Аннотация

В статье описаны две версии алгоритма контекстно-зависимого аннотирования документов, использованные компанией Яндекс для участия в семинаре РОМИП'2008. Алгоритмы основаны на формализации критериев качества, предъявляемых к аннотациям. Кроме описания самих алгоритмов, рассмотрены полученные результаты, а также обсуждена применимость этих алгоритмов для построения аннотаций на коллекциях промышленного масштаба.

1. Введение

При выполнении поиска в Интернете одним из важнейших факторов, определяющих качество ответа поисковой системы, является аннотация содержимого найденных документов, показываемая на странице результатов поиска. Для изучения возможностей улучшения этого аспекта поиска Яндекса мы реализовали два прототипа экспериментальной системы

аннотирования документов, описанные в данной статье. Работа над ними делилась на два этапа. На первом этапе было проведено неформальное изучение критериев качества, предъявляемых к аннотациям, и рассмотрены возможности их формализации. На втором этапе, на основании этих данных мы составили два экспериментальных варианта набора факторов для вычисления оценки релевантности и соответствующих им алгоритмов оценки качества, которые и использовали в нашей системе. По результатам экспертной оценки РОМИП [1] второй из вариантов оказался систематически превосходящим первый. Ниже описаны обе версии алгоритма и результаты сравнения их качества по данным РОМИП.

2. Описание первой версии алгоритма

Первый алгоритм построения аннотаций основан на вычислении нескольких признаков, которые описаны ниже, и, далее, на использовании известного приема упорядочения объектов (в нашем случае — предложений) при многокритериальной оптимизации.

Предполагается, что для каждого слова w документа d известны два числа: $\mathrm{IDF\_COL}(w)$ и $\mathrm{IDF\_DOC}(w,d)$, которые вычисляются по следующим формулам:

$$\mathrm{IDF\_COL}(w) = \log \frac{|D|}{|\{d_i \mid w \in d_i\}|},$$

$$\mathrm{IDF\_DOC}(w,d) = \log \frac{|P|}{|\{p_i \mid w \in p_i\}|},$$

где $|D|$ — количество документов в коллекции, $|\{d_i \mid w \in d_i\}|$ — количество документов коллекции, в которых встречается слово w, $|P|$ — количество абзацев (параграфов) в документе и $|\{p_i \mid w \in p_i\}|$ — количество абзацев в документе, в которых встречается слово w. В качестве коллекции документов для расчета $\mathrm{IDF\_COL}(w)$ использовалась коллекция русскоязычных документов, имеющихся в поисковом индексе Яндекса [2].

Будем использовать следующие обозначения: QL — множество лемм слов запроса, TL — множество лемм слов

заголовка документа, DL — множество всех лемм документа, $SL(s)$ — множество всех лемм предложения s, $\mathrm{TF}(w,d)$ — частота слова w в документе d, $\mathrm{DOC\_LEN}(d)$ — число слов в документе d и $\mathrm{AVG\_LEN}$ — среднее число слов в документе в коллекции.

Основная идея подхода данной версии алгоритма к учету множества признаков (не обязательно заранее определенного объема) и к ранжированию объектов s из множества d заключается в следующем известном приеме: минимизации расстояния между оцениваемым объектом s и некоторым «идеальным» объектом $perf\_s$, который характеризуется такими признаками:

$$f_i(perf\_s) = \max_{s \in d} f_i(s)$$

Здесь **max** f означает «наилучшее» значение признака f из всех имевшихся. Объекты упорядочиваются по возрастанию (евклидова) расстояния от объекта s до идеального объекта $perf\_s$.

Нами были посчитаны значения следующих восьми признаков для каждого предложения s документа:

1. $f_1(s) = -\sqrt{\sum_{w \in QL \setminus SL(s)} (\mathrm{IDF\_COL}(w))^2}$;
2. $f_2(s) = -\sqrt{\sum_{w \in TL \setminus SL(s)} (\mathrm{IDF\_COL}(w))^2}$;
3. $f_3(s) = -\sqrt{\sum_{w \in DL \setminus SL(s)} (\mathrm{IDF\_COL}(w))^2}$;
4. $f_4(s) = -\sqrt{\sum_{w \in QL \setminus SL(s)} (\mathrm{IDF\_DOC}(w))^2}$;
5. $f_5(s) = -\sqrt{\sum_{w \in TL \setminus SL(s)} (\mathrm{IDF\_DOC}(w))^2}$;
6. $f_6(s) = -\sqrt{\sum_{w \in DL \setminus SL(s)} (\mathrm{IDF\_DOC}(w))^2}$;
7. $f_7(s) = \sum_{w \in QL} \mathrm{IDF\_COL}(w) \times$

$$\times \frac{3 \cdot \mathrm{TF}(w,d)}{\mathrm{TF}(w,d) + 2 \cdot \left(0.25 + 0.75 \cdot \frac{\mathrm{DOC\_LEN}(d)}{\mathrm{AVG\_LEN}} \right)};$$

8. $$f_8(s) = \sum_{w \in QL} \mathrm{IDF\_COL}(w) \times$$

$$\times \frac{2.2 \cdot \mathrm{TF}(w,d)}{\mathrm{TF}(w,d) + 1.2 \cdot \left(0.4 + 0.6 \cdot \frac{\mathrm{DOC\_LEN}(d)}{\mathrm{AVG\_LEN}} \right)}.$$

Видно, что последние два признака являются реализациями широко известной формулы Okapi BM25 [3] с различными значениями свободных параметров. Что касается первых шести признаков, они представляют собой (евклидовы) расстояния до «идеального» («полного») набора лемм от фактического набора лемм предложения s в пространствах, соответственно, всех лемм запроса, заголовка документа и всего документа и в координатах, соответственно, $\mathrm{IDF\_COL}(w)$ и $\mathrm{IDF\_DOC}(w,d)$.

Перед вычислением весов предложений производилась нормировка вышеописанных признаков следующим образом. Для каждого признака были вычислены его математическое ожидание $\mathrm{M}(f_i)$ и дисперсия $\mathrm{D}(f_i)$ по всем предложениям документа. Далее, значения признаков нормировались по следующей формуле:

$$f_i(s) = \frac{f_i(s) - \mathrm{M}(f_i)}{\sqrt{\mathrm{D}(f_i)}}.$$

Окончательные веса предложений вычислялись также по вышеописанному принципу, как евклидово расстояние от предложения s до идеального предложения $perf\_s$, в пространстве восьми описанных выше нормированных признаков.

Согласно правилам дорожки контекстно-зависимого аннотирования длина аннотации не должна была превышать 300 символов. Поэтому алгоритм построения аннотации работает следующим образом. Берем предложение с минимальным весом, из тех, что еще не вошли в аннотацию. Если это предложение по числу символов влезает в текст аннотации, добавляем его туда. Если же не влезает, обрезаем его (причем так, чтобы его длина не превосходила

150 символов), и после обрезания добавляем в текст аннотации. Далее переходим к следующему по возрастанию веса предложению.

По завершении выбора, предложения, попавшие в аннотацию, сортируются в соответствии с порядком, с которым они шли в исходном документе. Если какие-нибудь два подряд идущих предложения аннотации не являются соседними в тексте документа, то они отделяются многоточием в тексте аннотации.

3. Описание второй версии алгоритма

Второй алгоритм построения аннотаций основан на вычислении нескольких признаков, описанных ниже, и лексикографической сортировке предложений по совокупности всех признаков.

Прежде чем рассмотреть признаки, введем понятие опорной пары предложения, позиции опорной пары и ширины опорной пары. Опорной парой предложения будем называть два наименее частотных слова (в смысле $\mathrm{IDF\_COL}(w)$) из пересечения слов запроса и предложения. Для однословных пересечений под парой имеется в виду единственное слово пересечения. Для предложений, в которых нет слов запроса, опорная пара не определена. Шириной опорной пары будем называть минимальное расстояние между словами опорной пары в предложении. Для однословных пересечений положим значение ширины равным размеру среднего предложения русского языка (10 слов). Позицией опорной пары будем называть позицию первого слова первой от начала предложения опорной пары минимальной ширины.

Под сегментом документа будем понимать функционально однородную область документа (меню, область контента, новостной блок, заголовок раздела и т.п.). Сегмент, в который попадает предложение, обозначим через $\mathrm{seg}(s)$. Сегменты будем разделять на информационные (например, область контента) и служебные (например, меню). Множество информационных сегментов будем обозначать через IS, а множество служебных сегментов через AS.

Рассмотрим теперь признаки, использованные для сортировки:

1. $f_1(s) = \sum_{w \in SL(s) \setminus QL} \mathrm{IDF\_COL}(w)$ — по убыванию;
2. $f_2(s) = \text{позиция опорной пары}$ — по возрастанию;

3. $f_3(s) = \text{ширина опорной пары}$ — по возрастанию;
4. $f_4(s) = \begin{cases} 1, & \text{если } s \text{ - в заголовке;} \\ 0, & \text{если } s \text{ - не в заголовке;} \end{cases}$ — по убыванию;
5. $f_5(s) = \begin{cases} 2, & \text{если } \text{seg}(s) \in IS; \\ 1, & \text{если } \text{seg}(s) \in AS; \end{cases}$ — по убыванию;
6. $f_6(s) = \sum_{w \in QL \setminus SL(s)} \text{IDF\_COL}(w)$ — по убыванию.

Приоритет признаков – от 6 к 1 (т.е., сортировка по шестому признаку, в случае равенства значений – по пятому, и т.д.). Алгоритм построения аннотации работает следующим образом. Берется первое предложение после сортировки. Если это предложение по числу символов влезает в текст аннотации, то добавляем его туда. Если же не влезает, то обрезаем его таким образом, чтобы длина его не превосходила 150 символов, а после обрезания добавляем в текст аннотации. Также добавляем все леммы предложения в так называемое множество использованных лемм. Далее производим еще одну пересортировку: смотрим, какие леммы запроса не вошли в множество использованных лемм, и переупорядочиваем предложения по убыванию суммы $\text{IDF\_COL}(w)$ слов запроса, не вошедших в первое предложение. После этой пересортировки продолжаем добавлять предложения (обрезая их по необходимости), пока очередное предложение влезает в разрешенную максимальную длину текста аннотации. При этом мы пропускаем предложение, если разность мощности множества лемм предложения и мощности множества использованных лемм меньше, чем четверть мощности множества лемм предложения. Последний прием позволяет избежать добавления предложений, содержащих только те слова запроса, которые уже вошли в предыдущие добавленные предложения.

Далее, предложения, попавшие в аннотацию, сортируются в соответствии с порядком, с которым они шли в исходном документе. Если какие-нибудь два подряд идущих предложения аннотации не являются соседними в тексте документа, то они отделяются многоточием в тексте аннотации.

4. Оценка результатов

Оценка результатов сделана в рамках РОМИП, следующим образом:

4.1 Получение оценок

Результаты оценивались только по коллекциям KM и BY. В итоге (на 29 сентября) было оценено (по две оценки на каждый результат) 60 запросов, и 1896 документов. Для каждого запроса было исходно отобрано для оценки 50 случайных документов (итак, средний уровень оцененности запроса составил чуть более 63%). Затем из оценки были исключены документы с пустым заголовком.

- каждый документ оценивался двумя асессорами
- для оценки предъявлялась следующая информация:
  - текст запроса
  - расширенное описание запроса
  - заголовок документа
  - тексты аннотаций (первые 300 символов текста)
- текст самого документа асессору не предъявлялся
- асессору давались сразу все доступные аннотации для данного документа по данному запросу, в случайном порядке
- асессор обязан выставить каждой аннотации две оценки:
  - оценку информативности
  - оценку читабельности
- и кроме того, ответить на следующие вопросы:
  - исходя из полученной информации, считаете ли вы, что документ содержит релевантную информацию? (ответы: да, нет, не могу сказать)
  - приняли ли бы вы такое же решение, используя только заголовок документа?
- оценки выставлялись по 9 градациям (1-9), которые были сгруппированы в 3 группы (1-3 – плохо, 4-6 – хорошо, 7-9 – отлично)

4.2 Метрики качества

Метрики качества вычислялись следующим образом:

- при вычислении метрик оценки трех групп (1-3, 4-6, 7-9) получали числовые значения 1, 2, 3 соответственно; различие оценок внутри группы не учитывалось

- из наличных оценок брались минимальная и максимальная (согласно описанной процедуре, задание давалось двум асессорам, так что и оценок было не более двух)
- оценки информативности и читабельности усреднялись по всем документам, отдельно по минимальной и отдельно по максимальной оценке (четыре значения на каждый прогон)
- также вычислялся и усредненный процент отличных оценок

Кроме того, были посчитаны усредненные оценки отдельно по релевантным, нерелевантным документам, и по документам, где релевантность, оцененная по аннотации, совпала с истинной релевантностью ответа (оцененной заранее) – в каждом случае в двух вариантах: оценивая релевантность по шкалам AND и OR. Всего 24 значения для каждой версии (в статье не приводятся). Результаты сравнения по этим оценкам не отличаются значительно от результатов сравнения по общему усреднению.

4.3 Полученные результаты

По усреднению оценок трех групп:

	Инф. (min)	Инф. (max)	Чит. (min)	Чит. (max)
Версия 1	2.549	2.769	1.951	2.418
Версия 2	2.598	2.793	2.133	2.572

По всем оценкам вторая версия устойчиво лучше первой. Та же картина сохраняется и по частичным усредненным оценкам.

Сравнение двух прогонов по усреднению оценок трех групп:

	информативность	читабельность
Вер2 vs Вер1	2.19%	12.93%

Приведены усредненные по оценкам min и max показатели, вычисленные следующим способом:

- из всех значений вычитается 1.0 (минимально возможное значение оценки)
- разность между вторым и первым прогоном делится на значения для второго прогона
- то есть, смысл этих чисел – «на сколько процентов первый прогон хуже второго»

По проценту отличных оценок:

	Инф. (min)	Инф. (max)	Чит. (min)	Чит. (max)
Версия 1	61.9%	80.8%	22.7%	47.4%
Версия 2	65.8%	82.9%	32.5%	60.8%

Разница в качестве между второй и первой версиями становится здесь еще более заметна:

	информативность	читабельность
Вер2 vs Вер1	4.24%	26.11%

4.4 Обсуждение результатов

В статье приведены результаты сравнения двух экспериментальных версий алгоритма построения аннотаций. Как набор признаков, так и метод их использования были в этих версиях различны. Несмотря на общий выигрыш по качеству второй версии, можно сделать вывод, что обе версии содержат элементы, полезные для дальнейшей работы в этой области. Как в одной, так и в другой версии не были использованы методы машинного обучения на размеченном людьми материале. Построение промышленной версии алгоритма требует более систематического учета имеющейся статистики. Настоящая работа – один из подготовительных шагов в этом направлении.

Литература

[1] ROMIP 2008. http://romip.ru/ru/2008/tracks/annotation.html

[2] *И. Сегалович, М. Маслов, Ю. Зеленков.* Цели и результаты программы научных стипендий Яндекса, 2005. http://company.yandex.ru/grant/2005/00_YANDEX.pdf

[3] *S. E. Robertson, S. Walker, S. Jones, M. M. Hancock-Beaulieu, and M. Gatford.* Okapi at trec-3. // In Proc. of the TREC-3, 1994

Yandex algorithms of contextual annotation at ROMIP 2008

Alexey Brodskiy, Ruslan Kovalev, Michael Lebedev, Dmitry Leshchiner,

Pavel Sushin

Yandex
{abrodskiy, velavokr, mlebedev, leshch,
psushin}@yandex-team.ru
Ilya Muchnik
Data Analysis School
muchnikilya@yahoo.com

The article describes two versions of algorithm for contextual annotation, used by Yandex at ROMIP 2008. The results of algorithms comparison and quality evaluation figures are given.