Яндекс на РОМИП-2010. Тестирование простой ранжирующей формулы.

© Сафронов Александр

Яндекс
alsafr@yandex-team.ru

Аннотация

Статья представляет собой отчет об участии в дорожках веб-поиска семинара РОМИП. Описывается ранжирующая формула, приводятся полученные результаты.

1. Введение

Для выполнения заданий семинара в этом году мы использовали «лёгкую» экспериментальную поисковую систему. Данная система не имеет непосредственного отношения к основному поиску Яндекса, её главное назначение – проверка работоспособности некоторых текстовых факторов. В частности, в этом году на РОМИПе нами был опробован фактор ранжирования на основе размера минимального окна со словами запроса.

2. Алгоритм ранжирования

2.1 Общая формула

Для ранжирования результатов поиска использовалась простая линейная формула.

$$Score(d,q) = \sum_i k_i * F_i(d,q)$$

где

$Score(d,q)$ – итоговый вес документа d по запросу q;

$F_i(d,q)$ – i-й фактор;

k_i – вес i-го фактора.

Список факторов, которые участвовали в формуле:

1. BM25 для полного текста документа ([3]);
2. BM25 заголовка документа;
3. BM25 начальной части документа;
4. Вес самой длинной непрерывной цепочки слов запроса в документе;
5. «Кучность» слов запроса в тексте документа на основе фактора, описанного в работе [2];
6. YMW. Фактор на основе размера минимального окна, включающего максимальное количество встречающихся в документе слов запроса. Подробнее этот фактор будет описан ниже.

Таким образом, ранжирующий алгоритм включал в себя только 6 факторов. Для коллекции BY настройка параметров формулы производилась вручную, без привлечения методов машинного обучения. Оптимизация производилась по метрике Average precision. Для коллекции KM отдельная настройка параметров не производилась, использовались параметры, оптимизированные для коллекции BY.

2.2 Фактор YMW

Фактор представляет собой эвристическую модификацию формулы 1.3 из работы [1]. Суть модификации состоит в добавлении в формулу множителя, масштабирующего значение фактора для документов, в которые входят не все слова запроса.

$$YMW(d,q) = \frac{\log(\alpha)}{\log(mw(d,n) - |n| + \alpha)} * \frac{S(n)}{S(q) + \beta * (S(q) - S(n))}$$

$$n = d \cap q$$

$$S(x) = \sum_{t \in x} idf(t)$$

где

d – документ;

q – запрос;

α – константа;

n – множество слов запроса q, встречающихся в документе d;

$|n|$ – количество слов запроса q, встречающихся в документе d;

$mw(d,n)$ – размер минимального «куска» текста, в котором встречаются все слова из n;

β – константа;
$S(n)$ – сумма весов слов запроса, встречающихся в документе;
$S(q)$ – сумма весов всех слов запроса.

3. Результаты

Мы принимали участие в дорожках ad hoc поиска по коллекциям BY.web и KM.ru.

Для коллекции BY оценка производилась на основе 550 запросов (из них 50 было взято из прошлогодней дорожки). Глубина пула была равна 20.

	Relevant-minus				Relevant-plus			
	Or		And		Or		And	
	MAP	P10	MAP	P10	MAP	P10	MAP	P10
y-1	0,26	0,14	0,22	0,09	0,18	0,07	0,12	0,04
y-2	0,41	0,49	0,39	0,38	0,34	0,29	0,26	0,16
y-3	0,42	0,50	0,39	0,37	0,35	0,30	0,28	0,18
y-4	**0,42**	**0,50**	**0,40**	**0,38**	**0,35**	**0,30**	**0,29**	**0,19**
x-5	0,32	0,41	0,27	0,28	0,26	0,21	0,19	0,11
x-6	0,22	0,37	0,18	0,24	0,16	0,19	0,10	0,08
x-7	0,30	0,42	0,27	0,32	0,25	0,25	0,21	0,16
x-8	0,34	0,43	0,31	0,34	0,26	0,25	0,24	0,14
x-9	0,35	0,44	0,32	0,34	0,28	0,26	0,26	0,16
x-10	0,33	0,44	0,32	0,34	0,25	0,26	0,22	0,15

Таблица 1. Коллекция BY, некоторые бинарные метрики.

	DCG@10	nDCG@10	pFound	ERR
y-1	0,629	0,096	0,111	0,064
y-2	2,788	0,420	0,283	0,185
y-3	2,864	0,430	0,287	0,189
y-4	**2,891**	**0,433**	**0,289**	**0,190**
x-5	2,155	0,340	0,238	0,152
x-6	1,824	0,271	0,209	0,128
x-7	2,443	0,345	0,246	0,160
x-8	2,495	0,364	0,255	0,167
x-9	2,582	0,372	0,259	0,171
x-10	2,524	0,368	0,256	0,168

Таблица 2. Коллекция BY, graded метрики.

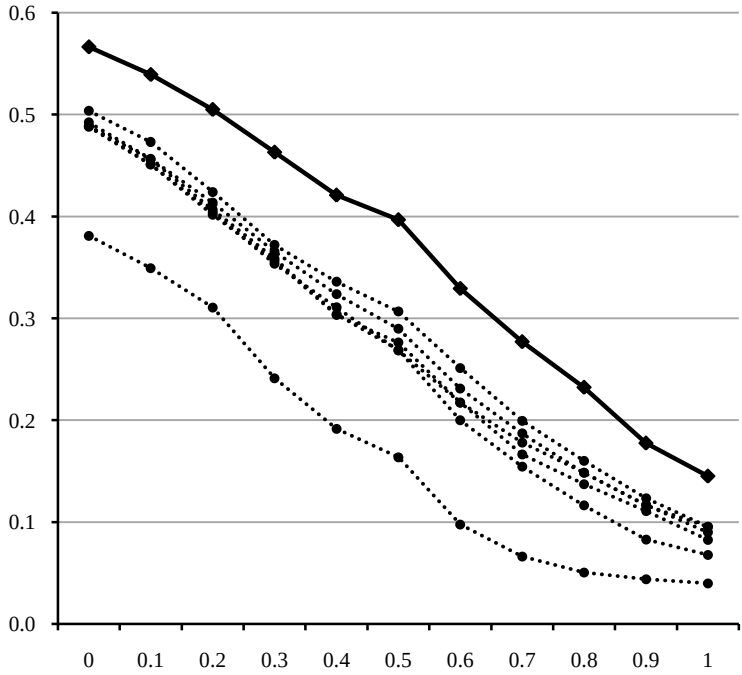

Рисунок 1. 11-точечный график TREC. Коллекция BY. Оценка relevant-plus-or. Один из прогонов Яндекса показан непрерывной линией, прогоны остальных участников – пунктиром.

Для коллекции KM всего было оценено 100 запросов, глубина пула составила 50 документов.

	Relevant-minus				Relevant-plus			
	Or		And		Or		And	
	MAP	P10	MAP	P10	MAP	P10	MAP	P10
y-1	0,51	**0,60**	0,48	0,51	**0,49**	0,43	**0,49**	**0,36**
y-2	**0,52**	**0,60**	**0,49**	**0,52**	0,47	0,43	0,47	0,36
x-3	0,46	0,53	0,43	0,47	0,43	0,41	0,44	0,32
x-4	0,34	0,48	0,35	0,45	0,36	0,39	0,40	0,33
x-5	0,45	0,56	0,47	0,52	0,45	0,44	0,45	0,35
x-6	0,45	0,57	0,47	0,52	0,45	0,45	0,46	0,34
x-7	0,44	0,56	0,45	0,50	0,45	**0,45**	0,47	0,35

Таблица 3. Коллекция KM, некоторые бинарные метрики.

	DCG@10	nDCG@10	pFound	ERR
y-1	5,687	0,513	**0,396**	0,298
y-2	5,646	**0,518**	0,393	0,290
x-3	5,228	0,455	0,350	0,261
x-4	5,085	0,399	0,330	0,255
x-5	5,579	0,496	0,381	0,290
x-6	5,721	0,504	0,388	0,302
x-7	**5,721**	0,490	0,382	**0,303**

Таблица 4. Коллекция KM, graded метрики. Оценки, вычисленные при использовании не более 50 первых ответов системы.

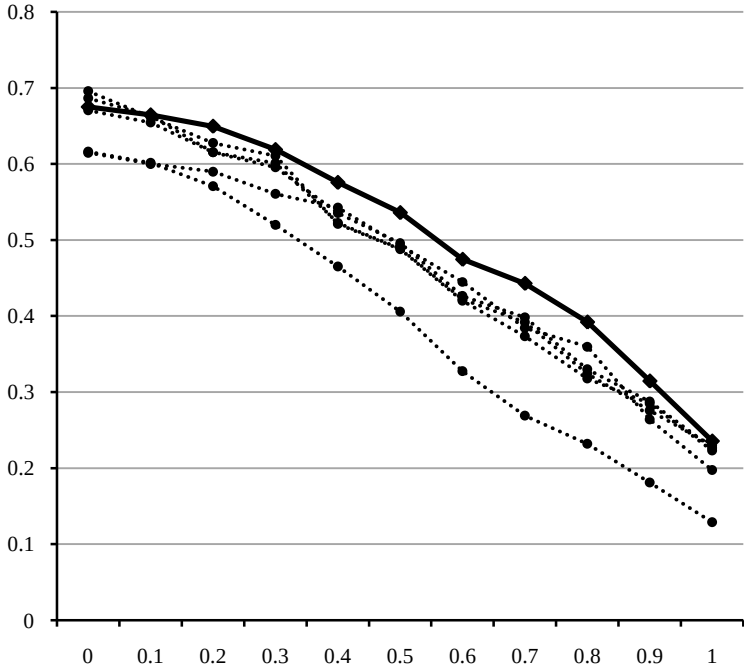

Рисунок 2. 11-точечный график TREC. Коллекция KM. Оценка relevant-plus-or. Один из прогонов Яндекса показан непрерывной линией, прогоны остальных участников – пунктиром.

4. Проверка нового фактора

Поскольку фактор YMW неплохо проявил себя в поисковых дорожках РОМИПа, было принято решение оценить пользу от его внедрения в основной поиск Яндекса.

Каждый новый фактор перед внедрением в основной поиск должен пройти ряд проверок, в том числе проверку на наличие в факторе полезного сигнала. Другими словами, необходимо получить объективное подтверждение того, что новый фактор действительно улучшает ранжирование.

В настоящее время в ранжирующей формуле Яндекса используется несколько сотен различных факторов. При этом для построения самой формулы используется машинное обучение (строго говоря, в данном случае правильнее говорить не о «формуле», а о полученной путем обучения ранжирующей модели). Большое количество параметров и сложность формулы делают задачу по оценке полезности нового фактора не совсем тривиальной.

Вероятно, самый простой способ проверки нового фактора состоит в том, чтобы разбить базу оцененных асессорами пар запрос/документ на обучающее и тестовое множество, после чего получить две ранжирующих модели – с тестируемым фактором и без него. Затем на тестовой выборке можно сравнить значение оптимизируемой метрики для этих моделей, и если модель, включающая новый фактор, дает улучшение метрики выше порогового, то считать этот фактор «хорошим».

Однако такой метод нельзя считать надежным по целому ряду причин. В частности, результат этой проверки сильно зависит от способа разбиения оцененной базы на обучающее и тестовое множество. При одном разбиении фактор может оказаться «хорошим», а при другом – «плохим». Кроме того, для алгоритмов машинного обучения, использующих bagging (т.е. взятие случайного подмножества обучающей выборки на каждой итерации обучения), характерны небольшие флуктуации качества получаемой модели, что также может негативно сказаться на стабильности оценки нового фактора.

Распространенной практикой решения подобных проблем является использование перекрестной проверки (она же – «скользящий контроль» или же «cross-validation»). Один из видов перекрестной проверки – k-fold cross-validation (k-кратный скользящий контроль). Он подразумевает, что база оцененных пар запрос/документ разбивается на обучающее и тестовое подмножество k разными способами. При этом размер тестовой

выборки составляет 1/k от всей базы, причем тестовые выборки не пересекаются. Для каждого разбиения («фолда») производится отдельное обучение и сравнение моделей.

Одна из процедур оценки нового фактора в Яндексе основывается на перекрестной проверке с последующим расчетом парного t-критерия Стьюдента. Минимальным количеством разбиений для перекрестной проверки считается 5, однако для надежной оценки «слабых» факторов (т.е. таких, которые улучшают ранжирование не очень заметно) может потребоваться проверка на нескольких десятках разбиений. Фактор считается хорошим, если в среднем на всех разбиениях он улучшает оптимизируемую метрику, и при этом с помощью t-критерия подтверждается гипотеза о неравенстве средних значений оптимизируемой метрики для моделей с фактором и без него.

Мы произвели оценку фактора YMW с помощью описанной выше процедуры. Тестирование производилось на 16 «фолдах», в качестве оптимизируемой метрики использовалась невязка.

№ разбиения	Невязка на тестовой выборке		Уменьшение невязки, %
	Без фактора	С фактором	
1	0,06287	0,06286	0,015
2	0,06149	0,06141	0,133
3	0,06148	0,06149	-0,022
4	0,06162	0,06160	0,041
5	0,06172	0,06167	0,082
6	0,06258	0,06252	0,082
7	0,06232	0,06226	0,104
8	0,06191	0,06188	0,047
9	0,06140	0,06138	0,042
10	0,06236	0,06236	0,008
11	0,06149	0,06146	0,054
12	0,06202	0,06192	0,170
13	0,06150	0,06143	0,111
14	0,06265	0,06253	0,197
15	0,06191	0,06183	0,131
16	0,06098	0,06092	0,094

Таблица 5. Тестирование фактора YMW на базе оцененных запросов Яндекса с помощью перекрестной проверки.

Значение t-критерия Стьюдента составило 0,0000776, что позволяет уверенно отвергнуть нулевую гипотезу. Поскольку при

этом использование фактора YMW уменьшает невязку в среднем на 0,08%, то можно сделать вывод о том, что этот фактор содержит полезный сигнал и будет улучшать ранжирование в основном поиске Яндекса.

Литература

[1] *М.С. Агеев, Б.В. Добров, Н.В. Лукашевич, А.В. Сидоров*. Экспериментальные алгоритмы поиска/классификации и сравнение с «basic line». Труды второго российского семинара по оценке методов информационного поиска. Стр. 62-89

[2] *А.В. Сафронов*. HeadHunter на РОМИП-2009. Российский семинар по Оценке Методов Информационного Поиска. Труды РОМИП 2009. Стр. 63-70.

[3] *S. Robertson, S. Walker, S. Jones, M. Hancock-Beaulieu, M. Gatford*. Okapi at TREC-3. In Proceedings of the Third Text REtrieval Conference (TREC 1994).