Фильтрация нерелевантных поисковых запросов при формировании семантического ядра сайта

Черникова Дарья Андреевна,
ООО «Рекламный агрегатор»

В рамках автоматизации подбора семантического ядра для сайта появился вопрос, как фильтровать запросы из разных источников. Базовая фильтрация, например, по количеству показов по этому запросу, не является достаточно интеллектуальной метрикой, она позволяет отфильтровать только совсем мусорные запросы. Сервисы для подбора семантического ядра часто отдают запросы, которые связанны с тематикой, но на самом сайте не освещены. Например, запросы «купить 3 д принтер» и «купить пластик для 3d принтера» для сайта, который оказывает услуги 3D-печати, но не продает принтеры и расходные материалы.

Для того, чтобы решить эту проблему, нужен фильтр по релевантности запроса сайту.

Исторически самым простым способом сделать такой фильтр была проверка наличия слов на странице. Для некоторых случаев это дает приемлемый результат, но для сложных случаев подход не работает. Если на сайте продаются стиральные машины, то слова из запроса «купить машину в москве» с большой вероятностью будут присутствовать на этой странице и запрос попадет в финальный список, что неправильно.

В качестве основной идеи был взят тот факт, что поисковые системы хорошо знают, по какому запросу какую информацию ожидает пользователь. Поэтому можно считать, что в выдаче по запросу представлены эталонные сайты. Если мы на них похожи, то запрос релевантен. Однако есть сайты, которые охватывают несколько тематик, одна из которых — наша, а значит, что не все запросы по этой тематике подойдут нашему сайту. Поэтому мы можем говорить про релевантность запросов только в разрезе конкретных страниц.

К тому же выдача поисковых систем часто содержит крупные агрегаторы, по некоторым запросам в выдачу подмешиваются форумы и т.д. Для того, чтобы сравнение была максимально корректным, был составлен список популярных доменов, страницы которых не участвуют в сравнении. Для получения этого списка получили выдачу по 40 тысячам запросов. После этого взяли страницы из топ-10, определили их домены, взяли в список 700 самых частотных. Среди них есть youtube.com, vk.com, ru.wikipedia.org, ru.aliexpress.com, pikabu.ru, kinopoisk.ru и т.д.

Чтобы отфильтровать форумы и любые другие сайты информационного характера реализовали алгоритм поиска телефона в контенте страницы. Практика показывает, что все коммерческие сайты указывают телефон для связи, поэтому удаляли страницы без указания телефона.

Для того чтобы страницы из выдачи было с чем сравнить, необходимо выбрать для запроса страницу на нашем сайте. В этом тоже может помочь поисковая система. Если использовать поиск по сайту site:{domain} {query} то в результате можно получить страницы с сайта {domain} наиболее релевантные запросу {query}.

Результат такого поиска дает неплохие результаты, но иногда лучшая страница находится не на 1 месте, а на 2 или 3. Поэтому лучше ориентироваться не на один запрос, а предварительно объединить их в логические группы. Для этого нужно найти пересечения в выдаче топ-10 по запросам, если пересечение страниц двух запросов в выдаче больше некоторого порогового значения, то эти запросы можно отнести к одной группе. В ходе экспериментов, наилучший

результат показал порог, равный 4.

После логической группировки можно выбрать наилучшую страницу для каждой группы по следующему алгоритму:

1. Для каждого запроса проверяем, нет ли в выдаче страницы нашего сайта на позиции меньше 50, если есть, то сохраняем эту страницу с найденной позицией.

2. Если на предыдущем шаге были найдены страницы, то для группы выбирается страница с минимальной позицией.

3. Если страниц в выдаче получено не было, то страница выбирается среди страниц, которые были получены поиском по сайту по каждому запросу, при этом при каждом поиске остается только 3 первых результата.

Для каждой страницы считаем:

Freq — количество раз, когда страница встречалась

AvgPos — средняя позиция, на которой страница встречается

На основании информации по всем страницам считаем:

Max(Freq) — максимальное количество раз, когда страница встречалась

Тогда показатель значимости страницы рассчитывается по формуле:

$$Importance_i = \frac{\text{Freq}_i}{\text{Max(Freq)}} * \frac{1}{\text{AvgPos}_i}$$

В качестве страницы для группы выбирается страницы с максимальным показателем значимости.

После того, как страницы для каждой группы были выбраны, можно приступить к сравнению со страницами из выдачи. При этом для сравнения берется 3 страницы, которые в выдаче по группе встречались чаще всего.

Для сравнения двух страниц было решено обучить модель, которая на основе запроса, контента нашей страницы и контента страницы конкурента определяла, является ли наша страница релевантной запросу.

Для ее обучения необходимо было найти положительные и отрицательные примеры. Нашли некоторый корпус запросов, по которым была получена выдача. Эти запросы отфильтровали по показам в месяц, чтобы не использовать для обучения низкочастотные запросы, т.к. для низкочастотных запросов страницы за пределами топ-10 часто нерелевантны.

Страницы на 20-30 позиции были выбраны для роли «наших страниц», по ним мы и будем оценивать релевантность. Страницы конкурентов брались из топ-10, при этом популярные домены и страницы без телефонов удалялись.

Так были подготовлены положительные примеры. В качестве отрицательных примеров для «наших страниц» выбирались страницы конкуренты по другим запросам. Это искусственный пример, который позволит обучить модель только очень простым случаям. Для того, чтобы получить сложные случаи, набрали огромный корпус запросов, среди которых находились пары, в которых один запрос являлся подмножеством другого.

Например:

· купить детское пальто — детское пальто

· купить машину — купить стиральную машину

Для этих пар получали выдачу, и в случае, если выдача была очень схожая, использовали эту пару для формирования положительного примера. При этом «нашу страницу» брали из одного запроса, а страницу конкурента из выдачи другого запроса. Если выдача была принципиально разной, то аналогичным образом фиксировали данные как отрицательный пример.

После отбора страниц для обучения, для них был получен контент. Контент и текст запроса нормализовали, а затем преобразовывали в вектор чисел с помощью TF-IDF. На базе этих векторов были построены следующие признаки:

1. Косинус между векторами нашего контента и контента конкурента.

2. Евклидово расстояние между векторами нашего контента и контента конкурента.

3. Манхэттенское расстояние между векторами нашего контента и контента конкурента.

4. Отношение количества пересекающихся биграмм (словосочетания из 2 слов) между биграммами нашего контента и биграммами контента конкурентов к сумме биграмм нашего контента и биграмм контента конкурентов.

5. Отношение количества пересекающихся частотных слов к сумме частотных слов нашего контента и частотных слов контента конкурентов. В список частотных слова берется топ-100.

6. Процент слов из запроса в контенте нашей страницы. Для расчета определяем количество слов из запроса, которое есть в контенте страницы и возвращаем отношение этого количества к количеству слов в запросе.

7. Процент слов из запроса в контенте страницы конкурентов. Для расчета определяем количество слов из запроса, которое есть в контенте страницы и возвращаем отношение этого количества к количеству слов в запросе.

8. Отношение процента слов из запроса в контенте страницы конкурентов к проценту слов из запроса в контенте нашей страницы.

9. Отношение слов в контенте нашей страницы, которые есть в запросе, к числу слов на этой странице.

10. Отношение слов в контенте страницы конкурента, которые есть в запросе, к числу слов на этой странице.

11. Среднее количество слов запроса в контенте нашей страницы. Для расчета считается количество вхождений каждого слова из запроса в текст страницы, а затем эти количества усредняются.

12. Среднее количество слов запроса в контенте страницы конкурента. Для расчета считается количество вхождений каждого слова из запроса в текст страницы, а затем эти количества усредняются.

13. Отношение среднего количества слов запроса в контенте страницы конкурента к среднему количеству слов запроса в контенте нашей страницы.

Для обучения использовался RandomForest, для проверки качества полученной модели использовалась стандартная кросс-валидация, а также была отложена небольшая часть выборки для проверки.

Результаты на отложенной выборке:

	precision	recall	f1-score	support
False	0.91	0.84	0.88	1278
True	0.94	0.97	0.96	3366
avg / total	0.93	0.93	0.93	4644

	False	True
False	1079	199
True	108	3258

Самыми значимыми признаками оказались:

· Манхэттенское растояние

· Процент слов из запроса в контенте нашей страницы

· Отношение количества пересекающихся биграмм между биграммами нашего контента и биграммами контента конкурентов к сумме биграмм нашего контента и биграмм контента конкурентов

После обучения модель можно использовать для оценки релевантности. Для каждого запроса в группе берется топ-3 страниц конкурентов. Формируются тройки «запрос-наша страница-страница конкурента» и отправляются для прогноза в модель. Так происходит для каждого запроса в группе, при этом страницы конкурента для всех одинаковые. Финальный ответ, является ли наша страница релевантной группе запросов определяется по большинству.

Модель показала хорошее качество на кросс-валидации, но, чтобы убедиться в ее работоспособности, еще один тест провели уже не на модельных данных, а на реальных. Для этого выбрали все запросы по проектам, которые пришли к нам за последние три месяца. Семантические ядра для этих запросов составляли люди. Для всех этих запросов мы подобрали страницу на сайте и проверили их релевантность. Основная идея была в том, чтобы убедиться, что модель не удаляет хорошие запросы. Опыт показал, что 95% запросов по нашей модели оказались релевантными.