Исследование значимых для ранжирования признаков в поисковой выдаче Google в развивающихся странах

Апухтин Дмитрий Игоревич,
ООО «Инструменты генерации дохода»

Поиском в интернете ежедневно пользуется множество людей. Решая их повседневные задачи, поисковые системы регулярно усложняют свой поиск. Итеративная доработка поиска приводит к тому, что факторов ранжирования становится так много, что они начинают учитываться не только в качестве самостоятельных единиц, но и как совокупность нескольких признаков. Из-за этого возникает проблема реверс-инжиниринга. Мы столкнулись с ней при разработке технологии APRA, предназначенной для формирования требований по доработке продвигаемой страницы и всего сайта с целью попадания в топ 10 органической выдачи поисковых систем.

Если предположить, что инвестиции разработчиков в поиск коррелируют с объёмом платёжеспособной аудитории, то алгоритмы ранжирования в менее развитых, с точки зрения интернета, странах, должны быть проще. Упрощённый алгоритм ранжирования позволит более точно определить какие именно факторы ранжирования существуют.

Для проверки гипотезы о зависимости сложности формулы ранжирования от развитости страны и использования результатов мы провели исследование сложности алгоритма Google в Иране и Таиланде.

Иран

Особенности языка

Официальный язык в Иране — персидский, он же фарси. Его особенности:

1. Направление письма — справа налево;

2. Отсутствие заглавных букв.

Первый пункт не важен для обработки текстов, а благодаря второму отсутствует необходимость приводить текст к единому регистру.

В русском языке есть служебные части речи, отсутствие которых делает текст неестественным. Для фарси есть аналогичный список служебных слов. Готовый список мы не нашли, поэтому пришлось составили его самостоятельно. Для этого потребовалось собрать тексты на фарси и найти инструмент для определения частей речи персидского языка. Мы выбрали библиотеку Hazm 0.5.2 на Python. Кроме того, в библиотеке есть методы для разбиения текста на слова и предложения, а также методы нормализации слов. Служебные части речи в этой библиотеке обозначаются так: P, CONJ, POSTP, Pe, INT. Собранные персидские тексты были разбиты на слова, а затем, с помощью библиотеки, определялись их части речи. Служебные части речи сохранялись. Затем был построен частотный словарь сохраненных слов, из которого затем удалили такие, частотность которых ниже порогового значения. Этот список в дальнейшем использовали для поиска в тексте служебных слов.

Сбор данных для исследования

Мы сформировали список запросов, по которому получили выдачу Google в Тегеране с помощью нашего внутреннего инструмента. Выдача представляет собой отсортированный по позициям список страниц по запросу. Контент всех страниц был скачан в формате html.

Генерация признаков и их разметка

Все признаки можно разделить на 2 типа: признаки страницы и признаки релевантности запросу. Признаки страницы отображают какие-то параметры страницы, например, абсолютные значения:

· количество символов на странице без учета html разметки;

· количество слов на странице;

· количество уникальных слов;

· частота самого частого слова.

А также признаки, прошедшие нормализацию к другому параметру:

· процент английских символов к общему числу символов;

· число слов к числу предложений;

· процент вариативности: отношение уникальных нормализованных слов к уникальным базовым словам;

· процент служебных слов на странице к общему числу слов;

· процент символов пунктуации к общему числу символов.

Признаки релевантности запросу рассчитываются по паре «Текст страницы — запрос», например:

· количество точных вхождений запроса в контент страницы, в meta, в заголовки h1-h3, в title;

· процент вхождений слов из запроса в контент страницы, в meta, в заголовки h1-h3, в title;

· покрытие title страницы триграммами из запроса;

· покрытие запроса триграммами из title страницы;

· индекс первого вхождения запроса на странице, нормированное на число символов.

Покрытие запроса триграммами title — один из самых сложных факторов, который рассчитывается следующим образом.

Скользящим окном формируются трехбуквенные сочетания (триграммы) для запроса и для title. В цикле перебираются все триграммы запроса, проверяется наличие этой триграммы в триграммах title, если такое совпадение существует, то из базового запроса удаляются символы, составляющие эту триграмму. На выходе получается отношение числа невычеркнуых символов к общему числу символов запроса. Аналогично рассчитывается покрытие title триграммами запроса.

Анализ полученных данных

Анализ проводился с помощью методов машинного обучения. На первой итерации обучения в качестве целевой функции, использовался класс позиции (0 класс — с 1 по 10 позицию, 1 класс — с 11 по 20 и т.д.). В качестве метода классификации использовалось дерево решений с глубиной 5. Такой простой метод был использован для того, чтобы в случае успеха, можно было интерпретировать полученные результаты и выделить значимые признаки. Стандартная кросс-валидация для этого случая не подходит, т.к. некоторые страницы появляются в выдаче многократно, и есть вероятность переобучиться. Например, в российской выдаче Википедия, Youtube и социальные сети часто занимают первые места. В иранской выдаче есть аналогичный шанс «запомнить» параметры хороших страниц и всегда прогнозировать им первый класс. Поэтому вместо стандартной кросс-валидации на каждой итерации обучения выбирались 85% из уникальных страниц. Для обучения использовались элементы из этого множества, остальные данные

использовались для проверки качества модели.

Ниже приведен пример матрицы неточностей и отчет классификации, где отображены полнота и точность каждого класса на одной из итераций обучения.

Матрица неточностей — это матрица размера N на N, где N — количество классов. Столбцы матрицы резервируются за правильными решениями, а строки — за решениями классификатора. Когда мы классифицируем объект из тестовой выборки мы инкрементируем число, стоящее на пересечении строки класса, который вернул классификатор, и столбца класса к которому действительно относится объект.

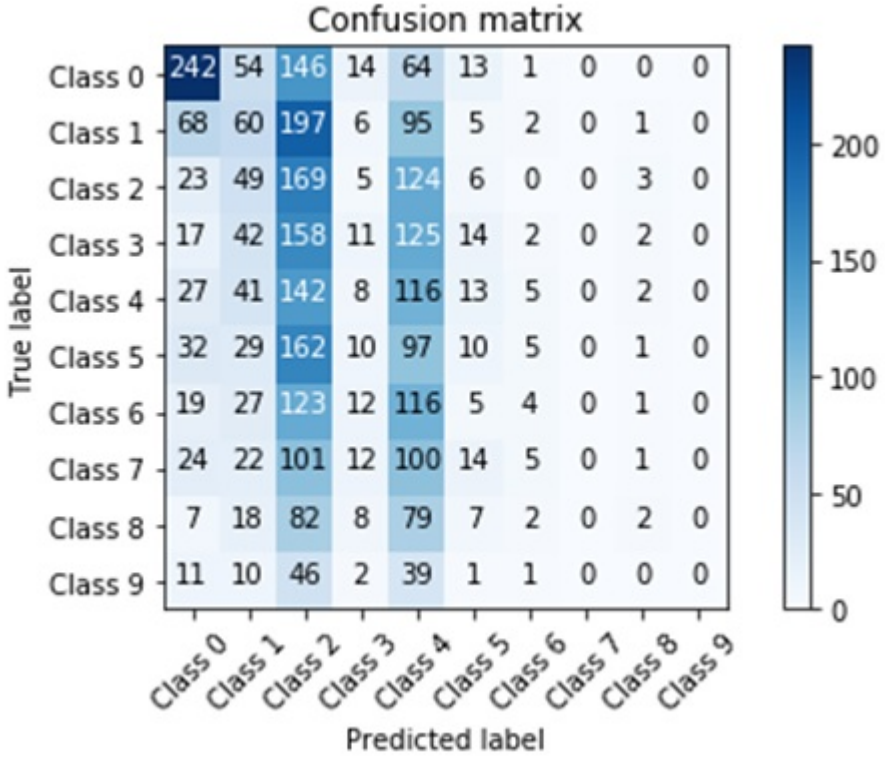
Отчет классификации выводится средствами библиотеки sklearn и отражает метрики полноты (recall) и точности (precision) для каждого класса.

Точность можно интерпретировать как долю объектов, названных классификатором положительными и при этом действительно являющимися положительными, а полнота показывает, какую долю объектов положительного класса из всех объектов положительного класса нашел алгоритм.

Для финальной оценки качества использовалась F-мера, которая представляет собой гармоническое среднее между точностью и полнотой. Она рассчитывается по формуле:

$$F_1 = 2\frac{Precesion \times Recall}{Precesion + Recall}$$

Далее приводятся примеры одной из итераций валидации, итоговая оценка классификатора всегда проводилась по усредненным значениям.



	precision	recall	f1-score	support
Class 0	0.51	0.45	0.48	534
Class 1	0.17	0.14	0.15	434
Class 2	0.13	0.45	0.20	379
Class 3	0.12	0.03	0.05	371
Class 4	0.12	0.33	0.18	354
Class 5	0.11	0.03	0.05	346
Class 6	0.15	0.01	0.02	307
Class 7	0.00	0.00	0.00	279
Class 8	0.15	0.01	0.02	205
Class 9	0.00	0.00	0.00	110
avg / total	0.18	0.18	0.15	3319

Можно заметить, что лучше всего классификатор отрабатывает на 0 классе, а классы с 5 по 9 имеют очень плохие показатели. Это может говорить о том, что класс 0 имеет какие-то ярко выраженные отличия, по которым его можно достаточно хорошо классифицировать, а также о том, что страницы ниже 10 позиции (класс 1-9) достаточно похожи, поэтому классификатор их не различает и определяет в классы 1-4.

Следующая итерация обучения основывается на выше описанных наблюдениях. На этом этапе мы превратили нашу задачу в задачу бинарной классификации: будем определять, страница находится в топ-10 или нет.

Обучение с помощью дерева решений дает следующие результаты:

	precision	recall	f1-score	support
Class 0	0.61	0.27	0.38	518
Class 1	0.87	0.97	0.92	2710
avg / total	0.83	0.86	0.83	3228

Обучение с помощью градиентного бустинга дает результаты чуть лучше:

	precision	recall	f1-score	support
Class 0	0.86	0.31	0.45	487
Class 1	0.87	0.99	0.93	2335
avg / total	0.87	0.87	0.84	2822

Видно, что для нулевого класса значение полноты небольшое, а также наблюдается соотношение классов 1:5, что дает некоторое смещение. Чтобы это исправить, уравняем количество объектов каждого класса, для этого оставим в первом классе столько случайно отобранных объектов, сколько есть в нулевом классе.

После подбора параметров для градиентного бустинга мы получили вариант, где полнота нулевого класса увеличилась до 0.5, но остальные показатели ухудшились.

	precision	recall	f1-score	support
Class 0	0.70	0.53	0.60	483
Class 1	0.72	0.85	0.78	703
avg / total	0.71	0.72	0.71	1186

Полученное качество классификатора не соответствует ожиданиям, поэтому на следующем этапе мы изучили исходные данные.

На этом этапе мы заметили:

1. Слова на фарси практически не изменяются, т. е. слова до и после нормализации не меняют своего написания. Такой вывод был сделан после анализа признаков, связанных с нормализованными вхождениями. Все эти признаки были исключены из обучения;

2. Первые три позиции в выдаче существенно отличаются (более чем в 3 раза) от остальных позиций по всем признакам, связанным с объемом страницы (количество символов на странице, количество слов на странице, количество уникальных слов).

Второе наблюдение подталкивает на мысль, что классифицировать нужно топ-3, а не топ-10, как мы это делали ранее. Но объектов с позициями 1-3 в наших данных всего 700, этого явно недостаточно для обучения. С точки зрения бизнеса, вывод в топ-3 гораздо полезнее, данные говорят о том, что результат должен получиться точнее, чем раньше, поэтому был осуществлен еще один сбор выдачи, но собирались только первые 3 страницы. Суммарно было размечено около 5000 страниц из топ-3. Количество объектов для обучения в классах было уравнено.

Обучение с помощью градиентного бустинга дало гораздо лучшие результаты, чем при обучении топ-10.

	precision	recall	f1-score	support
Class 0	0.81	0.83	0.82	969
Class 1	0.77	0.75	0.76	733
avg / total	0.80	0.80	0.80	1702

На текущий момент для обучения использовалось 46 признаков, классификатор на базе дерева решения давал результат намного хуже результата градиентного бустинга, поэтому не было возможности интерпретировать результаты и отсортировать признаки по значимости. На этом этапе мы провели удаление лишних признаков: обучали модель без одного признака, если среднее качество модели не ухудшалось в некотором диапазоне, то признак удалялся. После такой очистки у нас осталось 20 признаков:

· Признаки, характеризующие объем страницы (количество слов и символов на странице, количество уникальных слов, частота самого частого слова);

· Признаки, характеризующие естественность текста (процент символов пунктуации, процент служебных частей речи, процент английских символов на странице);

· Признаки вхождений слов запроса в различные части страницы (вхождения запроса в текст, факт наличия всех слов из запроса в тексте, процент слов из запроса в заголовках, мета тегах и т.д.)

Ранее обучение производилось с помощью библиотеки *sklearn*, следующим этапом стало использование другого инструмента для обучения — библиотеки *xgboost*. После перебора

параметров для обучения был получен следующий результат:

	precision	recall	f1-score	support
Class 0	0.88	0.90	0.89	880
Class 1	0.87	0.85	0.86	728
avg / total	0.88	0.88	0.88	1608

Параметры обучения:

n_estimators:55

learning_rate:0.4

min_child_weight:5

subsample:0.85

colsample_bytree:0.8

max_depth:7

eval_metric:'logloss'

objective:'binary:logistic'

В качестве вывода можно сказать, что нахождение страницы в топ-3 иранской выдачи практически на 90 % объясняется 20 признаками, характеризующими страницу и релевантность текста на странице запросу.

Таиланд

Следующая страна для анализа — Таиланд. Выбор обусловлен рядом особенностей языка, а также наличием open source библиотек для его анализа.

Таиланд относится к развивающимся странам, государственный язык — тайский. Особенности тайского языка:

1. отсутствие знаков препинания (точки, запятые, знаки вопроса и т.д.);

2. слова не разделяются пробелом;

3. предложения разделяются пробелом;

4. все слова имеют только одну форму: существительные не изменяются по падежам и не имеют родов, отсутствует форма множественного числа, нет разницы между прилагательными и наречиями, глаголы пишутся одинаково не зависимо от рода или времени;

5. строчные и прописные буквы не различаются;

6. порядок слов в предложении прямой: подлежащее — сказуемое — дополнение.

Все эти особенности учитывались при разметке признаков. Из-за отсутствия знаков препинания необходимо модифицировать разбиение на предложение. Не нужна нормализация и приведение текста к единому регистру.

Исследование значимых признаков осуществлялось по тому же алгоритму, который использовали в первом исследовании.

Из-за особенностей тайского языка потребовалась библиотека для разбиения текста на слова. В качестве такого инструмента использовалась библиотека PyThai 0.1.3 на языке Python. Так же был найден готовый список служебных слов в тайском языке, который содержит 112 слов.

Если для Ирана у нас были запросы, по которым можно было получить выдачу, то для Таиланда пришлось искать новый источник. В качестве источника использовался сервис Google Trends. Базовые 30 запросов были выбраны вручную из разных категорий в разделе популярных запросов, а затем каждый запрос расширялся запросами из колонки «похожие запросы». Новые запросы также проходили процедуру расширения до тех пор, пока колонка «похожие запросы» не была пуста.

По полученному списку запросов была получена выдача поисковой системы Google в регионе «Бангкок», для всех страниц из выдачи был скачан их контент в формате html.

На этапе генерации и разметки признаков мы адаптировали методы для разметки фарси, под особенности тайского языка, а также расширили их рядом новых признаков. Большинство из них касаются структуры страницы:

· количество заголовков h1-h3 на странице;

· количество ссылок на странице;

· длина title в символах;

· количество картинок на странице.

Также мы расширили признаки, отвечающие за естественность и сложность текста:

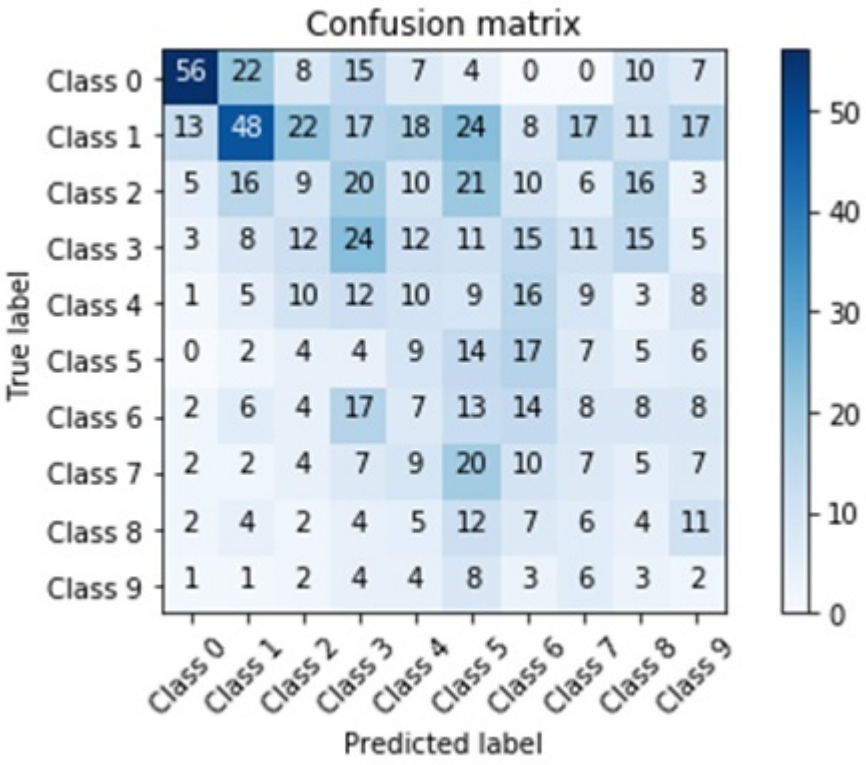
· процент коротких предложений к числу предложений;

· отношение количества предложений к числу символов;

· отношение числа уникальных предложений к общему числу предложений.

Прогнозирование, как и ранее, начнем к классификации на 10 классов (0 класс — с 1 по 10 позицию, 1 класс — с 11 по 20 и т.д.). Обучение проводилось на 11 тысячах примеров, а валидация на 1 тысяче. Кросс-валидация также осуществлялась описанным ранее способом. Для обучения использовалась библиотека *XGBoost*, т. к. она показала хорошие результаты.

Были получены следующие результаты:



	precision	recall	f1-score	support
Class 0	0.66	0.43	0.52	129
Class 1	0.42	0.25	0.31	195
Class 2	0.12	0.08	0.09	116
Class 3	0.19	0.21	0.20	116
Class 4	0.11	0.12	0.11	83
Class 5	0.10	0.21	0.14	68
Class 6	0.14	0.16	0.15	87
Class 7	0.09	0.10	0.09	73
Class 8	0.05	0.07	0.06	57
Class 9	0.03	0.06	0.04	34
avg / total	0.25	0.20	0.21	958

Классы 0 и 1 определяются лучше, чем остальные, результаты по остальным классам — приблизительно одинаковые, они близки к варианту классификатора, который случайным образом поставил бы метки классов от 0 до 9.

Чтобы определить порог для следующего этапа обучения, были внимательно были изучены исходные данные, но существенных отличий по среднему значению найдено не было, поэтому было сделано большое количество попыток обучения с подбором параметров и разным порогом разбиения на классы (от 5 до 20). Объемы классов всегда уравнивались. Результаты для разных разбиений после подбора параметров были похожи, но хуже всего себя показало разбиение от 5 до 7, вероятно, из-за самого маленького объема данных для обучения.

Результаты одной из итераций для разбиения на топ-10 и не топ-10:

	precision	recall	f1-score	support
Class 0	0.77	0.78	0.77	238
Class 1	0.64	0.63	0.63	148
avg / total	0.72	0.72	0.72	386

Параметры для обучения:

gamma:0.1

learning_rate:0.63

max_depth:8,

min_child_weight:4

n_estimators:60

subsample:0.8

colsample_bytree:0.8

objective:'binary:logistic'

eval_metric:"error"

Для проверки качества полученной модели, в процессе анализа была собрана дополнительная валидационная выборка. Эта выборка собиралась по запросам, которых не было при обучении, а также все страницы, которые участвовали в кросс-валидации, были из этой выборки исключены.

Результат представлен ниже:

	precision	recall	f1-score	support
Class 0	0.74	0.80	0.77	10798
Class 1	0.53	0.44	0.48	5508
avg / total	0.67	0.68	0.67	16306

Качество на валидационной выборке чуть хуже, чем на кросс-валидации, но в допустимых пределах. Так же можно увидеть, что нулевой класс (он соответствует страницам ниже 10 позиции) имеет лучший результат и при обучении, и при валидации. Результат по первому классу близок к классификатору, который случайным образом говорит, принадлежит ли объект к первому классу или нет, аналог подбрасывания монетки. Из этого можно сделать вывод, что признаки, гарантирующие, попадание в топ-10 нам выявить не удалось, но в алгоритмах ранжирования, скорее всего, есть факторы, значения которых мешают занимать высокие места. Примером такого фактора может быть отсутствие вхождений ключевого запроса в метатеги или в title. Именно такие зависимости мог найти классификатор, за счет чего и получилось высокое качество на нулевом классе. В случае, когда все признаки имеют допустимые значения параметров, т. е. поисковая система не пессимизирует страницу, в ранжировании принимают участие факторы, которых нет в нашем исследовании. Это могут быть, например, ссылочные или поведенческие факторы.

Отдельно исследовалась значимость всех признаков, по которым проводилось обучение. Сильно сократить число значимых признаков не удалось, т. к. качество прогноза значительно ухудшалось при удалении 3-4 признаков в разных комбинациях. Мы проанализировали значимость каждого признака с помощью встроенного в Xgboost метода. На каждой итерации первые два места по значимости делят между собой два признака: покрытие запроса триграммами из title и процент вхождений слов запроса в основной контент страницы. Следующие два значимых признака: процент вхождения в meta и в title. Остальные признаки сильно проигрывают по значимости.

Выводы:

По результатам исследования можно сделать следующие предположения:

1. Чем более развит интернет в стране, тем сложнее признаки, влияющие на ранжирование. В некоторых странах, например, в Иране, позицию в выдаче можно объяснить небольшим количеством признаков, характеризующими релевантность страницы запросу. В других странах, где поисковая система развивалась дольше, только текстовых признаков не хватает. Значения некоторых параметров могут мешать странице занимать высокие позиции в выдаче: например, отсутствие различных уровней заголовков или неестественно высокий процент длинных предложений. Возможно, такие правила работают, если по запросу за топ 10 конкурируют много сайтов, но если поисковой системе нечего показывать, и наша страница релевантна потребностям пользователя, то поисковая система покажет ее достаточно высоко.

2. Сложно сказать, насколько отличаются признаки для разных языков, можно только сказать, что логика разметки может сильно отличаться для разных языков или групп языков. Скорее всего существуют признаки, которые зарекомендовали себя как хорошие показатели релевантности и поисковые системы включают эти признаки в ранжирование большого количества стран. В нашем исследовании был найден как минимум один такой признак — покрытие запроса триграммами из title.

Наши исследования показали, что текстовые признаки вносят большой вклад в ранжирование только на начальных этапах развития поисковых систем в стране. Чем более развита страна, чем больше в ней активных пользователей интернета, тем сложнее алгоритм ранжирования, который учитывает текстовые признаки только для определения релевантности странице запросу, а когда

релевантных страниц находятся тысячи даже по низкочастотному запросу, то начинают учитываться другие признаки.