Алгоритм кластеризации поисковых запросов

Черникова Дарья Андреевна,
ООО «Рекламный агрегатор»

На первом этапе поискового продвижения формируется список запросов для продвижения. Их количество может достигать нескольких десятков тысяч. Дальнейшим шагом является объединение в группы (кластеры) одинаковых по смыслу запросов и выбор для каждой группы посадочной страницы на сайте. Вручную обработать такой список запросов очень сложно, поэтому в рамках разработанной нами технологии для привлечения клиентов из интернета Cubo потребовалось реализовать автоматическую кластеризацию.

Кластеризация — задача разбиения объектов на подмножества (кластеры), так, чтобы внутри одного кластера находились схожие объекты, а объекты в разных кластерах должны отличаться между собой.

Решение задачи кластеризации можно свести к следующим этапам:

1. определение характеристик (признаков) каждого объекта;

2. вычисление меры сходства между объектами;

3. применение алгоритмов кластеризации.

В качестве метрики сходства мы выбрали схожесть запросов по смыслу. Для ее использования нужно сначала преобразовать запросы в векторное представление признаков. Например, преобразовать каждый запрос в вектор TF-IDF.

TF-IDF — статистическая мера, используемая для оценки важности слова в контексте документа, который является частью корпуса или коллекции документов.

TF — частота слова, позволяет оценить важность слова в пределах одного документа.

$$tf(t,d) = \frac{n_t}{n_d}$$

n_t— число вхождений слова t в документ d;

n_d — общее число слов в документе;

IDF — обратная частота документа.

$$idf(t,D) = log\frac{|D|}{|\{d_i \in D | t \in d_i\}|}$$

$|D|$ — число документов в корпусе;

$|\{d_i \in D | t \in d_i\}|$ — число документов в корпусе D, в которых встречается слово t.

$$\text{tf} - \text{idf}(t,d,D) = tf(t,d) \times idf(t,D)$$

TF всегда рассчитывается в рамках одного документа (текста, запроса), а величина IDF — на основе всего множества документов (корпуса). Можно говорить о глобальном IDF для русского языка, если собрать достаточно большое количество текстов на русском, а затем по этим текстам посчитать IDF для каждого слова. Полученные величины будут отражать, насколько тот или иной

термин популярен в языке. Но для кластеризации запросов глобальный IDF не подойдет, т. к. нам важны не общетематические слова, вроде «дом», «машина», а слова, релевантные тематике запросов. Поэтому IDF нужно считать по всем запросам в семантическом ядре, а корпус будет состоять из списка всех запросов.

Наш алгоритм кластеризации выглядит так:

1. все слова из запросов нормализуются, удаляются служебные части речи (предлоги, союзы и т. д.);

2. каждый запрос преобразуется в вектор чисел с помощью TfidfVectorizer в библиотеке sklearn;

3. полученные векторы кластеризуются с помощью метода MiniBatchKMeans из библиотеки sklearn. Mini Batch K-Means является вариацией алгоритма K-Means, который использует мини-партии для сокращения времени вычисления, при этом работает незначительно потерями в качестве, относительно стандартного алгоритма.

Стоит отметить, что кластеризация — это задача обучения без учителя, тут нет правильных или неправильных ответов. Есть метрики, которые позволяют оценить качество кластеризации, но нам было важно, чтобы результат устраивал клиента, поэтому качество кластеризации оценивалось людьми. Результат кластеризации при каждом запуске может быть разный, т. к. при каждом запуске алгоритма в качестве начальных состояний используются случайные величины. Это не является проблемой, т. к. в реальности происходит то же самое: разные люди могут разбить семантическое ядро совсем по-разному. Кроме того, специалист в процессе работы с инструментом, может сделать перерасчет, если ему по каким-то причинам не понравился результат.

Примеры кластеризации одного семантического ядра:

— купить цветы германия — цветы с доставкой германия	— цветы с доставкой франция — купить цветы франция	— цветы с доставкой по москве — цветы на заказ москва

— купить цветы германия — купить цветы франция	— цветы с доставкой германия — цветы с доставкой по москве — цветы с доставкой франция	— цветы на заказ москва

В описанном подходе с использованием меры TF-IDF есть несколько проблем:

1. Специалист или пользователь инструмента кластеризации должен ввести количество кластеров. Для маленьких семантических ядер можно подобрать оптимальное число кластеров достаточно быстро, тем более, если специалист сам подбирал эти запросы, то он знает приблизительное количество групп. Но когда число запросов превышает несколько тысяч, итеративно увеличивать количество кластеров, пока не получится удовлетворяющий результат, очень долго, т. к. требуется время, чтобы оценить качество каждого кластера.

2. Синонимы и близкие по смыслу слова могут оказаться в одной группе в очень редких случаях, когда их TF-IDF будет очень похож в рамках семантического ядра. А хотелось бы, чтобы такие слова объединялись в группы как можно чаще.

В примере ниже, при разбиении списка запросов на две группы, гарнитура оказалась в одном кластере с домашним кинотеатром, хотя логически это неправильно.

— блютуз наушники — наушники цена — наушники сони — купить наушники в москве	— домашний кинотеатр 2.1 — домашний кинотеатр сони — домашний кинотеатр цена — беспроводной домашний кинотеатр — купить гарнитуру для телефона — гарнитура для телефона — гарнитура самсунг

Чтобы научить алгоритм находить похожие слова по смыслу, а не по частоте их употребления (вторая проблема), вместо TF-IDF, можно использовать Word2Vec. Этот инструмент был разработан Google в 2013 году. Алгоритм обучается на большом объеме текста, а затем его можно использовать для получения векторного представления слова. При этом векторные представления близких по смыслу слов будут похожи. Именно это позволяет находить синонимы, сокращения и т.д.

Для тестирования мы решили не обучать новую модель, а взять готовую. Обученная модель была взята на сайте https://zenodo.org/record/400631**.** Слова в этой модели не были нормализованы, каждое слово представляет собой вектор из 300 элементов.

Каждый запрос состоит из нескольких слов, для кластеризации все запросы необходимо преобразовать в векторы одной длины. Для этого можно представить каждый запрос как сумму векторов слов, которые его составляют.

Формула для сложения N векторов:

$$\overrightarrow{a_i} = \{a_{1i}; a_{2i}..; a_{ni}\}$$

$$\sum_{i=1}^{M} \overrightarrow{a_i} = \{\sum_{i=1}^{M} a_{1i}; \sum_{i=1}^{M} a_{2i} ...; \sum_{i=1}^{M} a_{ni}\}$$

n — размерность вектора $\overrightarrow{a_i}$;

M — количество векторов для сложения.

Векторное представление слов по Word2Vec не предполагает, что какие-то операции над векторами могут дать ценный результат, так как сами вектора не несут полезной информации о слове, смысл имеет лишь расстояние между векторами.

Но рассмотрим пример, когда у нас есть 3 запроса:

· купить машину;

· купить автомобиль;

· купить квартиру.

Векторы слов «машина» и «автомобиль» будут более схожи, чем векторы слов «машина» и «квартира», поэтому при суммировании мы получим векторное представление, по которому запрос «купить квартиру» будет отличаться от двух других запросов достаточно сильно, чтобы выделить его в отдельный кластер. Очевидно, что полученное векторное представление для запроса «купить машину» может быть получено суммированием нескольких других слов, которые по тематике очень далеки от автомобиля, но этим можно пренебречь, т. к. обычно семантическое ядро состоит из тематически связанных запросов, в которых нет такого разнообразия лексики, которое создавало бы случаи объединения совершенно непохожих запросов в одну группу.

Ниже примеры кластеризации семантического ядра с использованием Word2Vec:

— заказать цветы китай купить цветы чехия — цветы с доставкой германия	— заказать букет маме — заказать цветы мужчине — заказать цветы невесте — цветы девушке с доставкой — цветы учителю на дом	— анемоны дешево — гиацинты с доставкой — ландыши на дом — лилии недорого — мимоза с доставкой — розовые розы на заказ — розы гран при недорого — тюльпаны заказать

В этом примере получились 3 группы:

· группа запросов со странами;

· группа запросов про букеты кому-то в подарок;

· группа запросов с названиями цветов.

Задача кластеризации на этом этапе решена достаточно успешно. Дополнительное требование к инструменту — автоматическое назначение названий для групп.

Название группы может представлять собой список слов или словосочетаний. Все запросы в группе должны быть покрыты хотя бы одним словом или словосочетанием из списка. Определение названия происходит итеративно.

1. Определяется самое частое слово в группе, если с ним связано другое слово, то формируем словосочетание. Слово или словосочетание, которое мы получили, добавляем в финальный список названия;

2. Вычеркиваем из группы все запросы, которые содержат слово или словосочетание из первого пункта;

3. Повторяем пункты 1-2, пока в группе не закончатся запросы.

Пример кластеризации запросов с названиями групп:

доставка срочный	**экспресс доставка**	**грузоперевозка, авиаперевозка**	**экспресс доставка**
— срочная доставка владивосток — срочная доставка документов по россии — срочная доставка москва новороссийск — международная срочная доставка	— экспресс доставка в австралию — экспресс доставка в индию — экспресс доставка во вьетнам — экспресс доставка в грецию	— стоимость авиаперевозки грузов — стоимость жд грузоперевозок — транспортные грузоперевозки цена грузовые авиаперевозки цена — морские грузоперевозки цена	— экспресс доставка москва калининград — экспресс доставка сергиев посад — экспресс доставка документов нижний новгород — экспресс доставка в новороссийск из москвы

Из примера видно, что две группы имеют одинаковое название, но одна про доставку внутри страны, а другая про международную доставку. Человек легко подберет правильное название для каждой из групп, но сделать это автоматически нашим методом не получится.

Проблема определения оптимального числа кластеров (проблема № 1) относится к нерешенным проблемам кластерного анализа. Поэтому для каждой конкретной задачи подбираются эвристики, которые дают хоть какой-то результат.

Одно из возможных решений проблемы — найти зависимость между числом кластеров и количеством запросов.

Недостаток: не учитываются возможные различия между семантическими ядрами, например, у нас может быть, как мелкий клиент с большим количеством групп, так и большой клиент со всего несколькими тематиками.

Другое решение — подобрать допустимый диапазон расстояний между векторами. Можно увеличивать количество кластеров, если среднее расстояние между векторами больше максимального значения диапазона и уменьшать, если среднее расстояние меньше минимального значения. Проблема такого подхода в том, что есть клиенты, у которых только один товар, и его кластеры будут является разновидностями этого товара. Попытка склеивания этих групп до допустимого диапазона приведет к формированию одного кластера. А есть клиенты, у которых интернет-магазин, и на одной странице бытовая техника, а на другой — ювелирные изделия. В таком случае расстояние будет очень большим и разбиение до допустимого диапазона даст очень большое число кластеров.

Для другого решения можно использовать название групп, а точнее, количество слов и словосочетаний, которые это название формируют. Интуитивно понятно, что чем меньше слов или словосочетаний присутствует в названии группы, тем больше похожи запросы внутри нее. Если среднее число слов и словосочетаний в семантическом ядре очень большое, это значит, что внутри каждой группы собралось очень много тем, предметов или услуг. В таком случае количество кластеров нужно увеличивать. При этом, если требовать, чтобы каждое название содержало ровно одно слово или словосочетание, большое количество кластеров приведет к тому, что синонимичные понятия не будут объединяться в одну группу. Поэтому мы экспериментально подобрали коэффициент, при достижении которого можно больше не увеличивать число кластеров. Этот коэффициент равен 1.3. Подобный подход работает достаточно хорошо для любых семантических ядер.

Результат кластеризации с автоматическим определением числа кластеров:

Название кластера	**Запросы**
оборудование	клининговое оборудование
	моющее оборудование
	оборудование для клининга
	уборочное оборудование
высокий давление аппарат	аренда аппарата высокого давления
	ремонт аппарата высокого давления
керхер минимойка	купить минимойки керхер
	купить минимойку керхер
	минимойка керхер купить
	минимойка керхер цена
	минимойки керхер цена
	ремонт минимойки
	ремонт подметальной машины

ремонт,сервис	ремонт поломоечной машины
	ремонт пылесосов
	сервис поломоечной машины
	сервис пылесосов
высокое давление	аппарат высокого давления
	аппарат высокого давления купить
	аппарат высокого давления отзывы
	аппарат высокого давления цена
	аппараты высокого давления
	аппараты высокого давления купить
	аппараты высокого давления отзывы
	аппараты высокого давления цена
	купить аппарат высокого давления
	купить минимойку высокого давления
	минимойка высокого давления
	минимойки высокого давления
	минимойки высокого давления купить
минимойка, пароочиститель	купить минимойку
	купить пароочиститель
	минимойка купить
	минимойка отзывы
	минимойка цена
	минимойки купить
	минимойки отзывы
	минимойки цена
	пароочистители купить
	пароочиститель для дома
	пароочиститель купить
	пароочиститель цена
аренда	аренда коммунальной техники
	аренда минимойки
	аренда подметальной машины
	аренда поломоечной машины
	аренда пылесоса

	аренда пылесосов
	аренда уборочной техники
поломойка	ремонт поломойки
	сервис поломойки

Общий алгоритм кластеризации:

1. Очистка и нормализация запросов;

2. Получение векторного представления запросов с помощью Word2Vec;

3. Кластеризация полученных векторов с заданным числом кластеров (по умолчанию = 3);

4. Получение названий всех кластеров;

5. Расчет среднего значения количества слов и словосочетаний в названиях кластеров. Если полученное значение больше 1.3, то возврат на пункт 3 с изменением числа кластеров (Шаг увеличения = 3);

6. Склейка исходных форм с базовой (одной нормальной форме может соответствовать несколько исходных) и вывод пользователю результатов.