Векторное представление слов – вектор в пространстве с фиксированной размерностью. На вход подаются коллекция документов, на выходе получаем векторное представление из коллекции документов.

One-hot encoding – метод представления в векторном виде. Создается словарь фиксированного размера *N*, где для каждого слова из словаря соответствует вектор размера *N*, каждый вектор состоит из 1 единицы на -м месте и (*N – 1*) нулей, где – номер слова в словаре. Таким образом, запись данного метода представлена на рисунке 1.1.

ТУТ ДОЛЖЕН БЫТЬ РИСУНОК

Рисунок 1.1 – One-hot encoding

Из приведённом описание и примере можно сделать вывод, что данный метод обладает множеством недостатков:

* 1. Размерность векторов могут быть довольно большой;
  2. Слова, не попавшие в словарь, не будут обработаны;
  3. Векторы слов не отражают смысл слов;
  4. Семантическая значимость слов между собой отсутствует.

Bag to Words (BoG) – Мешок слов, представление предложений, основанный на методе one-hot encoding. В данном случае вектором предложений является сумма векторов слов. Или же можно сказать, что мешок слов для предложения – это вектор, у которого длина равна длине словаря и на -м месте стоит такое число сколько раз -е слово из словаря встречается в данном предложении. Запись данного метода можно увидеть на рисунке 1.2.

ТУТ ДОЛЖЕН БЫТЬ РИСУНОК

Рисунок 1.2 – Bag to Words

Следовательно, данный метод наследуется от предыдущего, будет иметь такие же недостатки, но только уже для предложений:

1. Предложения, не попавшие в словарь, не будут обработаны;
2. Векторы предложений не отражают смысл слов;
3. Семантическая значимость предложений между собой отсутствует.

И для того, чтобы перейти к следующему методу, стоит обратить внимание на еще один недостаток – различные слова в любом тексте имеют разную важность/вес для передачи контекста и смысла этого текста. Например, любой текст на английском языке имеет множество артиклей, предлогов, которые в свою очередь не несут никакого смыслового контекста, т.к. мы все же сможем понять без них, о чем идет речь. На рисунке 1.3 красным выделено, какие слова в каждом предложение имеют высокую важность для понимания, зеленным те, которые не имеют высокой важности.

ТУТ ДОЛЖЕН БЫТЬ РИСУНОК

Рисунок 1.3 – Важность слов в предложениях

Тогда перейдем к методу, который учитывает данную важность и будет упомянут RAG. TF-IDF – вычисляет важность каждого слова в документе относительно количества его употреблений в данном документе и во всей коллекции текстов. Этот метод позволяет выделить ключевые слова и понять, какие слова имеют больший вес для определенного документа в контексте всей коллекции. Или же более простыми словами – мера важности слова для документа среди документов .

Алгебраическая форма записи выгляди следующим образом:

(1.1)

где – входное слово;

– текущий документ;

– коллекция документов;

– относительная частота встречаемого слова в документе;

– обратная частота встречаемого слова в наборе документов.

Относительная частота рассчитывается следующим образом:

(1.2)

где – входное слово;

– документ, в котором ищется слово ;

– количество слова в документе ;

– количество слов в документе .

Обратная частота рассчитывается следующим образом:

(1.3)

где – входное слово;

– коллекция документов, их количество;

– количество документов, где слово встречается.

Следовательно, значение обратной частоты показывает, чем оно больше, т.е. количество слов встречается реже во всех документах, тем оно является более важным по смыслу в текущем документе, в противном случае, как было сказано про артикли и предлоги до этого, их количество велико, что говорит нам о том, что их важность очень мала и не несет никакой смысловой нагрузки.

Теперь, когда можно найти важность каждого слова среди документов, строятся векторы документов из данных значений, который имеет размер словаря *N* и на -й позиции вектора будет стоять значение -го слова данного документа. Для наглядности пример можно увидеть на рисунках 1.4 – 1.5.

ТУТ ДОЛЖЕН БЫТЬ РИСУНОК

Рисунок 1.4 – Матрица TF-IDF

ТУТ ДОЛЖЕН БЫТЬ РИСУНОК

Рисунок 1.5 – Векторное представление TF-IDF

Тем самым, данный метод добавляет важные характеристики:

1. Векторы документов приобретают смысл;
2. Ранжирование и выделение ключевых слов в документах.

Среди недостатков можно выделить следующие:

1. Размерность словаря фиксирована;
2. Размерность векторов могут быть довольно большой;
3. При изменении документов требуется перерасчет значений.

На основе предыдущего метода была разработана модификация TF – IDF – Best Maching 25 (BM25) – полнотекстовый поиск заключается в нахождении *наиболее релевантных* запросу документов из множества вариантов.

Алгебраическая форма записи выгляди следующим образом:

|  |  |
| --- | --- |
|  | (1.4) |

где: – документ;

– запрос;

– обратная частота документа;

– частота слова в документе.

Часто слова в документе выглядит следующем образом:

|  |  |
| --- | --- |
|  | (1.5) |

где – запрос;

– документ;

– частота запроса в документе;

– свободные коэффициенты (, , );

– длина документа;

– средняя длина документа в нашей коллекции.

Обратная частота документа выгляди следующим образом:

|  |  |
| --- | --- |
|  | (1.6) |

где – общее количество элементов в коллекции;

– количество документов, содержащих элемент запроса;

– это количество документов, не содержащих элемент запроса.

Сам алгоритм BM25, как и было выше сказано на принципе вероятностного ранжирования – Если извлечённые документы упорядочены по уменьшению вероятности релевантности данным, то эффективность системы оптимальна для этих данных.

К сожалению, вычислить истинную вероятность релевантности документа запросу почти невозможно.

Поэтому, дошли до того, что можно упростить уравнение за счет того, что важен порядок документов, точнее вес документа, а не точная вероятность. Можно убрать те члены уравнения, которые были необходимы для вычислений вероятности там, где их важность ничего не изменит с учетом существующего порядка.

Уравнение вычисляет вес на основании частотности элементов:

|  |  |
| --- | --- |
|  | (1.7) |

где – это вес рассматриваемого документа;

– это вероятность того, что элемент запроса встретится в документе с заданной частотой , если документ релевантен

Получаем, что уравнение сводится к тому, что можно увидеть частотность элементов запроса в документе, если документ релевантен или не релевантен, и вероятности того, что элемент совсем не встретится, если документ релевантен или не релевантен.

Следовательно, данный модифицированный метод добавляет нам то, что теперь можно выбирать ключевую информацию из релевантных документов, что в разы повышает точность ответа.

С основными методами нахождения ранжированных слов, предложений и документов разобрались. Теперь необходимо как-то сформировать верное сформулированное и понятное по смыслу предложение. Одна из моделей – N-граммы – это последовательность из N подряд идущих символов, звуков, слов, слогов и т.п. в тексте. Данная модель может быть:

1. Униграммой (1-граммы) – отдельные символы, слова и т.п.;
2. Биграммой (2-граммы) – пары слов;
3. Триграммы (3-граммы) – тройки слов.

Статическая модель, предсказывающая или предугадывающая следущее слово, основанное на вероятностной модели. Рассчитывается по цепному правилу:

|  |  |
| --- | --- |
|  | (1.8) |

где – слово в тексте;

– общий случай N-граммы.

Как можем заметить, что вероятность слова зависит от предыдущего , а не от всего предыдущего текста, что было до этого, тогда общий случай выглядит следующим образом:

|  |  |
| --- | --- |
|  | (1.9) |

https://habr.com/ru/articles/675218/

http://www.machinelearning.ru/wiki/images/b/b3/word2vec.pdf