Сборка словаря со-встречаемостей в bigARTM

Михаил Солоткий

Московский государственный университет им. М.В.Ломоносова

Факультет вычислительной математики и кибернетики

11 июня 2018 г.

Что хочется рассказать...

Постановка задачи

- Имеется 2 текстовых файла: коллекция в формате vowpal wabbit и словарь валидных токенов в формате UCI vocab.
- Параметры алгоритма: ширина окна, минимальная со-встречаемость документов и токенов.
- На выходе получить 4 файла в одинаковом формате: в каждом записаны строки текста в виде: token_id token_id value
- Обрабатывается только базовая модальность
- Среди 4 файлов 2 файла со-встречаемостей: документной и абсолютной и 2 файла со значениями ppmi, посчитанными по этим со-встречаемостям

Ширина окна

- Окно шириной k внутри документа - это k токенов слева от данного и k токенов справа. Если данный токен находится слишком близко к началу или концу документа, то берутся токены вплоть до границы документа.
- Пример при $k = 2$:

  Шла Саша по шоссе и сосала сушку

  |@verb Шла |@default_class Саша по шоссе и |@verb сосала |@default_class сушку

Подробнее о выходных данных

- Побочный эффект моего алгоритма: все данные в выходных файлах отсортированы.
- В файлах со-встречаемостей есть симметричные пары, в файлах ppmi - нет.

Документная и абсолютная со-встречаемости

- Абсолютная со-встречаемость - количество раз, сколько данная пара встретилась в коллекции в окне заданной ширины - не количество окон!
- Документная со-встречаемость - количество документов, в которых хотя бы раз встретилась данная пара в окне заданной ширины.

Как считались ppmi

- $ppmi(u, v) = \left[log \frac{p(u, v)}{p(u)p(v)} \right]_+ = \left[log \frac{n_{uv} n}{n_u n_v} \right]_+$
- Для абсолютной со-встречаемости n - количество всевозможных пар в тексте, которые находятся в окне заданной ширины.

$$n_u = \sum_v n_{uv}$$

- Для документной со-встречаемости n - число документов в коллекции
- n_u - количество документов, в которых встретилась данная пара

Как запускать

- bigartm -c vw -v vocab –cooc-min-tf 200 –cooc-min-df 200 –cooc-window 10 –write-cooc-tf cooc_tf –write-cooc-df cooc_df –write-ppmi-tf ppmi_tf –write-ppmi-df ppmi_df
- Пометки о запуске добавлены в BigARTM Command Line Utility и Python Guide » Tokens Co-occurrence and Coherence Computation

Шаг 1: чтение и составление батчей со-встречаемостей

- Загрузка токенов из vocab в хэш-таблицу. Здесь каждому токену присваивается порядковый номер начиная с 1.
- Документы коллекции читаюся некоторыми порциями в оперативную память, парсятся на токены.
- Происходит проход по токенам, берутся пары, попавшие в окно и которые есть в vocab.
- Факт их со-стречаемости запоминается, то есть пара "записывается"в красно-черное.
- Содержимое дерева записывается в выходной файл в так называемый "Батч со-встречаемостей"
- Берётся новая порция документов, и всё это многопоточно.

Шаг 2: агрегация по батчам

- Открываются все или, если их слишком много, почти все батчи, из них читаются ячейки, которые в каждом батче отсортированы по номеру первого токена.
- На ячейках создаётся приоритетная очередь.
- Ячейка сливается с промежуточным буфером, а из промежуточного буфера готовые данные могут быть записаны в оканчательный выходной файл.

Шаг 3: подсчёт ppmi

- Читаются выходные файлы со-встречаемостей, по ним считаются значения ppmi