Домашнее задание 2

В данном домашнем задании Вам предстоит научится быстро находить кандидаты на неточные дубли в коллекции документов, реализовав алгоритмы, описанные в лекции и предложив свои улучшения на основе знаний о природе коллекции документов.

Для оценки качества разработанного алгоритма Вам будут предоставлены результаты работы «тяжелого» алгоритма при попарном сравнения документов на дубли.

0. Коллекция

Для экспериментов Вам предоставляется коллекция документов с исходными кодами программ взятых из одной известной системы автоматического тестирования.

Каждый документ коллекции представлен в виде txt файла, в неизвестной кодировке, с преобладанием латинских символов. Документы даны как есть, без специальной структуры. Ознакомьтесь, пожалуйста, с содержимым нескольких из них, чтобы понять природу их происхождения. Архив с документами можно не распаковывать, а читать напрямую из кода. Пример чтения архива на питоне – в ноутбуке рядом с заданием.

1. Точные дубли

Прочитайте документы и проведите простейшую нормализацию содержимого документов, которая не меняет его сути. Например, следует сделать как минимум: разбор кодировки, удаление BOM, замена табуляций и переводов строк на пробелы, удаление последовательностей пробельных символов. Но только пожалуйста никаких, прости господи, лемматизаций и стемминга.

Рассчитайте свою любимую многобитную хеш-функцию для нормализованных документов и найдите точные дубли с помощью известной структуры данных.

Оцените какие группы дубликатов получилось найти: количество групп, минимальный, максимальный и средний размер.

Оставьте из каждой группы один документ для дальнейших заданий.

2. Ground truth

В файле *ground_truth.tsv.bz2* содержится результат работы «тяжелого» алгоритма сравнения документов на дубли для всех пар документов. Результат для одной пары документов — это число от 0 до 1. Отсутствие пары в архиве означает, что их схожесть равна нулю. Далее будем называть содержимое этого файла - *ground truth*.

Пример чтения архива на питоне – в ноутбуке рядом с заданием. Для иноязычных, далее в этом параграфе идет описание формата файла. Каждая строка файла описывает список потенциальных дубликатов для одного из документов коллекции. Строка файла разбита на колонки символом табуляции. В первой колонке записано имя документа, для которого далее следует список дублей. Далее в каждой колонке записано имя файла и оценка задублированности через символ = (равно).

Для каждой пары документов, которые Вы посчитали дублем в пункте 1, убедитесь, что данная пара есть в ground truth и что значение равно единице (aka полные дубли).

Так как точные дубли не представляют интереса для поиска неточных дублей исключите из ground truth пары, найденные в пункте 1.

3. MinHash

Разбейте документ на слова с учетом специфики содержимого документов: разделения по пробелам может быть недостаточно. Также учтите, что пунктуационные «слова» не менее важны в этой задаче, чем буквенные.

Рассчитайте MinHash описанный на лекции для всех документов. Разбейте документы по дублям из расчета, что дубли имеют одинаковый MinHash.

Реализуйте поиск ближайших документов для произвольного количества MinHash хеш-функций (k). Отсортируйте документы кандидаты по убыванию степени «похожести» – доле совпавших хешей из k.

Оцените качество подбора дубликатов при различных значениях k. Например, для k равных 1, 10, 50, 100. Для оценки качества воспользуйтесь списком ground truth. В качестве метрики качества предлагается использовать nDCG@10 где релевантностью выступает оценка задублированности из ground truth. Для кандидатов, которых нет в ground truth оценку задублированности стоит считать равной нулю. Пример расчёта метрики на питоне – в ноутбуке рядом с заданием.

Реализованный алгоритм поиска должен быть заметно быстрее, чем линейный проход по всем документам коллекции.

Дополнительно

4. SimHash (+1 балл)

По аналогии с пунктом 3, реализуйте предложенный на лекции алгоритм Random Hyperplane SimHash. Если алгоритм реализован правильно, то суммарное количество ноликов и единичек в хешах должно быть примерно одинаковое.

Степень «похожести» двух документов в данном случае будет доля совпавших бит в хеше. Для поиска ближайших по расстоянию Хэмминга хешей используйте приём с разбиением хеша на части.

Оцените качество подбора дубликатов для различных размеров хеша (n). Например, для n равных 16, 64, 128, 256. Для оценки качества используйте ту же метрику, что и в пункте 3.

5. Расширенная нормализация (+0.5 балла за пункт)

5.1. Предложите и реализуйте «нормализацию» слов документа, которая устойчива к переименованию именованных сущностей документа. Проверьте на сколько увеличилась полнота срабатывания пункта 1, а также качество пунктов 3 и 4 с новой нормализацией.

5.2. Предложите и реализуйте удаление слов документа, которые не влияют на логику документа. Проверьте на сколько увеличилась полнота срабатывания пункта 1, а также качество пунктов 3 и 4 с новой нормализацией.