Анализ неструктурированных данных

Домашняя работа 1

Векторные представления слов, тематическое моделирование, классификация текстов.

Дедлайн: 24.10.2018

Результатом выполнения задания является ipython-ноутбук, в котором представлены все скрипты, реализующие проделанные эксперименты и отчет, описывающий каждый шаг и всю логику вычислений и все полученные результаты. Отчёт также должен содержать краткое описание всех использованных моделей (с указанием на цитируемый источник описания модели, если такой имеется) и инструментов. Вся проделанная работа должна быть понятна из этого текста.

На усмотрение проверяющего остается штраф (т.е. снижение оценки) за неаккуратное оформление, копирование Википедии и любого другого ресурса без указания источника, списывание, неформальный стиль изложения и обилие стилистических ошибок.

Домашнее задание выполняется в группах по 1-3 человека.

Все вопросы по содержанию, оформлению и сдаче домашнего задания можно задавать учебным ассистентам в чате АНД 2018.

Выполненное домашнее задаение сдается через систему AnyTask. Инструкции по использованию системы AnyTask будут дополнительно опубликованы в телеграм-канале.

Часть I (6 баллов) В этой части задания задания требуется решить задачу классификации текстовой коллекции EUR-lex, используя различные признаковые пространства.

Формулировка задания:

1. **(1 балл)** Предобработка данных. Требуется привести данные в подходящий для работы вид, применить знакомые вам средства предобработки. Все проделанные операции подробно описать в отчёте.

2. **(2 балла)** На основе имеющихся данных подготовить с помощью изученных в курсе инструментов **минимум четыре** признаковых пространства, описывающих датасет. Обязательно должны быть признаки, полученные с помощью word2vec/FastText/Glove, тематического моделирования и doc2vec. Выбор реализаций инструментов свободный.
   Примечание: Реализацию doc2vec лучше всего использовать из пакета gensim, а именно модель DBOW с параметром `dbow_words=0`.

3. **(2 балла)** Выбрать один или несколько алгоритмов классификации (на свой вкус) и применить их к полученным признакам, оценивая качество классификации. Провести сравнение результатов: какой алгоритм классификации сработал лучше и почему? Как параметры признаковых пространств (например, размерность эмбеддинга) влияют на качество работы алгоритма классификации?

Примечание: алгоритм классификации стоит выбирать так, чтобы он позволял посчитать указанные ниже метрики качества, которые предлагается оптимизировать.

4. **(1 балл)** Запустить на предобработанных данных FastText и сравнивать полученное качество с результатами предыдущего пункта (код для подсчёта указанных в задании метрик качества нужно написать самостоятельно).

Датасет EUR-lex предлагается в виде двух файлов, `eurlex_data.txt` и `eurlex_labels.txt`. Файл с данными содержит два столбца, первый — идентификатор документа, второй — сам документ. Файл с метками состоит из трёх столбцов, первый — имя метки класса, второй — документ, к которому эта метка относится (у каждого документа может быть несколько меток классов), третий — константа 1, которую можно игнорировать. Выборку следует разбить случайным образом на две части в соотношении 9:1. Большая часть должна быть использована для обучения, меньшая — для тестирования.

Ссылка на данные: dropbox

Метриками качества будут ROC AUC и Precision-Recall AUC (PR AUC), посчитанные на тестовой выборке. Будем считать эти величины следующим образом: для каждого документа будем рассматривать вектор, длиной в число меток классов. В векторе ответов будут 1 на позициях верных классов, и 0 - на остальных позициях. В векторе предсказаний на каждой позиции будет вероятность данной метки в данном документе. Будем рассчитывать AUC для указанной пары векторов, после чего просуммируем и усредним получившиеся значения по всем тестовым документам. Для подсчёта обоих типов AUC для одного документа рекомендуется использовать реализации метрик из `sklearn`.

В обязательном порядке требуется написать код, демонстрирующий истинные и предсказанные метки классов и включить в отчёт примеры его работы.

Часть II (4 балла) Важно не только классифицировать данные, но и грамотно их визуализировать.

В этой части требуется визуализировать векторы, полученные с помощью word2vec, и матрицу «слова-темы», полученную с помощью тематического моделирования. Выбор инструментов построения модели и визуализации свободный. Требуется, чтобы визуализация была наглядной, давала какую-то информацию о данных и была, по возможности, красивой. Рекомендуется смотреть в сторону t-SNE и библиотек визуализации графов и тематических моделей (самый простой вариант — LDAvis). Пример визуализации матрицы «слова-темы» можно найти здесь. Визуализации должны быть представлены в отчёте в ipython-ноутбук вместе с подробным описанием того, как они были получены и какие выводы были сделаны на их основании.