Домашнее задание №2
по моделям LDA и word2vec

Срок сдачи задания: 18.05.2015

В рамках данного задания предлагается применить методы тематического моделирования (LDA) и нейросетевую модель (word2vec) к коллекции англоязычных отзывов фильмов. Предполагается использование языка *python* и библиотеки *gensim*. Результаты присылайте в виде ipython-notebook (код + результаты работы моделей + ваши выводы) на почту: *anya_potapenko@mail.ru* и *dmitrii.ignatov@gmail.com*. Тема письма должна иметь вид **[CL-ML2015-HW2] Фамилия Имя**, в теле письма укажите, какие пункты Вы выполнили.

Обязательная часть задания (10 баллов):

1. Зарегистрироваться на *kaggle.com*, посмотреть контест *"Bag of Words Meets Bags of Popcorn"* и скачать данные (100 000 отзывов о фильмах):

   www.kaggle.com/c/word2vec-nlp-tutorial/data.

2. Список библиотек, которые потребуются для выполнения задания:

   www.kaggle.com/c/word2vec-nlp-tutorial/details/setting-up-your-system

   Самый быстрый способ получить нужный набор (для windows):

   - Скачать Anaconda: store.continuum.io/cshop/anaconda/ и запустить exe-файл, который установит сразу много полезных пакетов (Python 2.7 + numpy + scipy + IPython + NLTK + ...).
   - В командной строке (cmd.exe) доустановить gensim: *conda install gensim*.

3. Внимательно разобраться с тьюториалами, доступными в рамках контеста (Part 1 – Part 4). Они очень подробные и полезные для понимания принципов анализа текстовых данных в Питоне.

4. Согласно тьюториалам подготовить данные и обучить модель word2vec. Полезно засечь время обучения модели *(from time import time)*.

5. Протестировать модель на датасете, подготовленном Google, с задачами вида: London → England, значит, Berlin → Germany. Пример вызова нужной функции и ссылка на датасет есть в подробном тьюториале по word2vec: radimrehurek.com/2014/02/word2vec-tutorial/. С чем может быть связано большое число неверных ответов?

6. Привести по 5-10 примеров для задачи поиска лишнего слова *(model.doesnt_match)* и для задачи поиска семантически близких слов *(model.most_similar)*. Проинтерпретировать полученные результаты. Все ли примеры соответсвуют здравому смыслу? По каким принципам группируются близкие слова в выбранных вами примерах?

7. Провести лемматизацию текстов с помощью пакета NLTK, подготовить частоты встречаемости слов в документах и обучить модель LDA на 50-100 темах (из пакета gensim). Вывести *время построения* и *перплексию модели* при выбранных параметрах. При выполнении этого пункта полезно опираться на демонстрацию с семинара. Также можно почитать тьюториал про LDA на Википедии: `https://radimrehurek.com/gensim/wiki.html`.

8. Вывести списки топ-слов для всех тем, вручную выбрать несколько примеров хорошо и плохо интерпретируемых тем (критерий – можете ли вы лаконично назвать тему или сформулировать принцип, по которому в нее собраны слова). Проинтерпретировать полученные результаты. Как вы думаете, какие особенности коллекции могли повлиять на качество результата?

Бонусная часть задания:

1. По коллекции построены две интересные модели – LDA и word2vec. Подсчет автоматических мер качества и анализ топ-слов / близких и лишних слов для нескольких примеров не дает полного представления о качестве и структуре модели. Предложите и реализуйте *любой дополнительный способ визуализации (анализа)* полученных моделей.

   Например, для word2vec это может быть так:

   - Снизить размерность скрытого слоя до 2. С этим хорошо справляется метод t-SNE. Готовую реализацию можно взять из библиотеки scikit-learn (sklearn.manifold.TSNE) или использовать более быстрый приближенный аналог Barnes-Hut t-SNE.
   - Нарисовать точки (слова) на плоскости, возможно, не все, а случайную подвыборку. При этом сделать так, чтобы при наведении курсора возникала подпись слова (это можно сделать средствами библиотеки matplotlib: `http://matplotlib.org/examples/event_handling/pick_event_demo.html`). Правда ли, что семантически близкие слова группируются вместе?

   Для тематических моделей существуют готовые средства визуализации, например, `http://vis.stanford.edu/papers/termite`. Можно пробовать разбираться.

2. Контест, данными которого мы пользуемся, посвящен анализу тональности текстов. По отзыву фильма нужно предсказать, положительный он или отрицательный. Попробуйте решить эту задачу любым известным вам методом и загрузить в систему Kaggle для дальнейшей оценки. Большинство классификаторов машинного обучения можно брать готовыми из библиотеки scikit-learn, к которой написано множество подробных тьюториалов. Какого качества удалось добиться в среди списка лидеров?