Группа 5: Макунина А., Суворова М., Рубанова Е., Берсенева М.

(1)

В проекте использовались: неразмеченный корпус из 19000 отзывов, LdaModel из библиотеки gensim, Adagram, WordNet, word2vec (обученный на Википедии и обученный на корпусе отзывов).

(2)

С помощью **AdaGram** и **WordNet** получили слова-соседей и синонимы для каждого из двух начальных списков оценочных слов категорий Food и Service. Далее обработали полученные списки, убрали случайно попавшие в выборки слова, избавились от повторов, а также разметили тональность слов вручную. В итоге у нас получилось 156 слов темы еда и 248 слов темы обслуживание. В этих корпусах содержатся как оценочные, так и тематические слова, принадлежащие разным частям речи.

Следующий шаг - **topic modeling**. Для этого запустили **LdaModel** из библиотеки **gensim** на целом корпусе из 19000 отзывов, в результате чего получили 4 тематических группы из 10 слов. Взяли только две из них, которые лучше остальных описывали слова категорий Food и Service.

Затем определяли, к какой категории принадлежит каждое слово из отзыва, попробовали два варианта:

1) обучили **word2vec** на корпусе из 1500 отзывах из корпуса; сравнивали similarity слов из отзыва со словами из топиков.
2) взяли уже обученную (на текстах Википедии) модель word2vec и сравнивали вектор слова из отзыва с вектором топика.

Второй способ показал лучший performance. Он использовался для определения аспекта слов далее.

На следующем этапе определялась тональность слова из отзыва (или относящегося к нему слова). Здесь мы тоже испытали два алгоритма:

1) ищет сами слова из сидов в тексте;
2) ищет близкие слова к словам из топиков и сидов

Первый алгоритм работает просто и **ищет слова из расширенных нами тональных списко**в, обрабатывая случаи типа "не вкусный".

Второй алгоритм **оценивает близость слов** из отзыва к **векторам топиков Food и Service,** определяя таким образом аспект, а также **близость к векторам Positive и Negative**, сделанных на основе расширенных списков тональных слов.

Дальнейшая обработка слов зависит от частеречной принадлежности. В случае наречий мы сразу сравнивали близости с нашими векторами, а вот для прилагательных и существительных необходимо было также смотреть на главное/зависимое слово, которое для простоты было определено позиционно. Мы пытались использовать разметку UDPipe для работы с синтаксическими отношениями, но её структура и не самое лучшее качество не позволили улучшить работу алгоритма. Мы думаем, что найдя удачное решение для проблемы поиска главного или зависимого слова можно значительно улучшить качество определения тональности лексикона.

В целом, первый алгоритм работает **точнее** и **лучше находит** тональную лексику (известную нам по спискам), но при этом второй, за счет нахождения близких слов, **охватывает более широкий диапазон лексики**. Итоговая функция извлечения тональных слов **объединяет два описанных алгоритма** и выдает информацию о тональной лексике, использованной в отзыве, в формате, сходном нашей ручной разметке фрагмента корпуса.

В итоговом варианте мы не использовали размеченные с помощью UDPipe файлы и высчитывали индексы слов вручную, что привело к несовпадению порядковых индексов слов в предложении. Оно было вызвано разницей в процессе токенизации, использованном нами и алгоритмом UDPipe. Эту проблему нам исправить не удалось.

Программа проекта находится в 4 файлах:

1) seed_extension.ipynb - расширение списков оценочных слов; способ с word2vec, обученном на корпусе отзывов.
2) topic modeling.ipynb - получение тематических групп с помощью LdaModel.
3) pretreatment.ipynb - преобразование списков и сохранение их в удобном виде.
4) classifier.ipynb - финальный классификатор слов из отзывов; способ с предобученным word2vec.

(3)

Протестировать алгоритм не представляется возможным за неимением золотого стандарта для всего корпуса.