Отчёт по лабораторной работе по курсу ММАТ
«Векторные представления слов, тематическое моделирование. Анализ тональности.»

Аят Оспанов
517 гр., ММП, ВМК МГУ, Москва
8 мая 2017 г.

Содержание

1 Постановка задачи **1**
1.1 Классификация текстовой коллекции EUR-lex 1
1.2 Визуализация признакового пространста 2
1.3 Анализ тональности текстов постов Twitter 2

2 Классификация текстовой коллекции EUR-lex **2**
2.1 Предобработка данных 2
2.2 Создание признакового пространства 4
2.3 Классификация 4
2.4 Сравнение предсказаний 5

3 Визуализация признакового пространста **6**
3.1 Word2Vec 6
3.2 LDA 7

4 Анализ тональности текстов постов Twitter **7**
4.1 Предобработка данных 7
4.2 Создание признакового пространства 8
4.3 Классификация 8
4.4 Визуализация 9

5 Заключение **10**

1 Постановка задачи

1.1 Классификация текстовой коллекции EUR-lex

Дана коллекция документов EUR-lex. Датасет EUR-lex предлагается в виде двух файлов, `eurlex_data.txt` и `eurlex_labels.txt`. Файл с данными содержит два столбца, пер-

вый – 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.

Для каждого документа из датасета нужно определить классы, к которым этот документ будет относится. Для этого надо сделать следующие шаги:

- Предобработка данных
- Создание признакового пространства, описывающее датасет
- Классифицировать датасет с помощью алгоритмов классификаций
- Посчитать метрики качества ROC AUC и PR AUC

1.2 Визуализация признакового пространста

В этом пункте нужно визуализировать векторы, полученные с помощью word2vec, и матрицу «слова-темы», полученную с помощью тематического моделирования. Требуется, чтобы визуализация была наглядной, давала какую-то информацию о данных и была, по возможности, красивой.

1.3 Анализ тональности текстов постов Twitter

Дан датасет постов из Twitter. Датасет содержит слудющие столбцы:

- ItemID – id документа
- Sentiment – класс (0 или 1)
- SentimentSource – ресурс, откуда был взят датасет
- SentimentText – текст документа.

2 Классификация текстовой коллекции EUR-lex

2.1 Предобработка данных

Данные из EUR-lex содержат уже обработанные данные представленные в виде токенов. Поэтому особой обработки над текстами не проводилось. Была сделана лишь следующая обработка:

- Были удалены стоп-слова. Стоп-слова были взяты из библиотеки `nltk` (`nltk.stem.snowball.stopwords`). Были взяты все языки, которые были в библиотеке, т.к. язык документов не один и определению нами не подлежит.

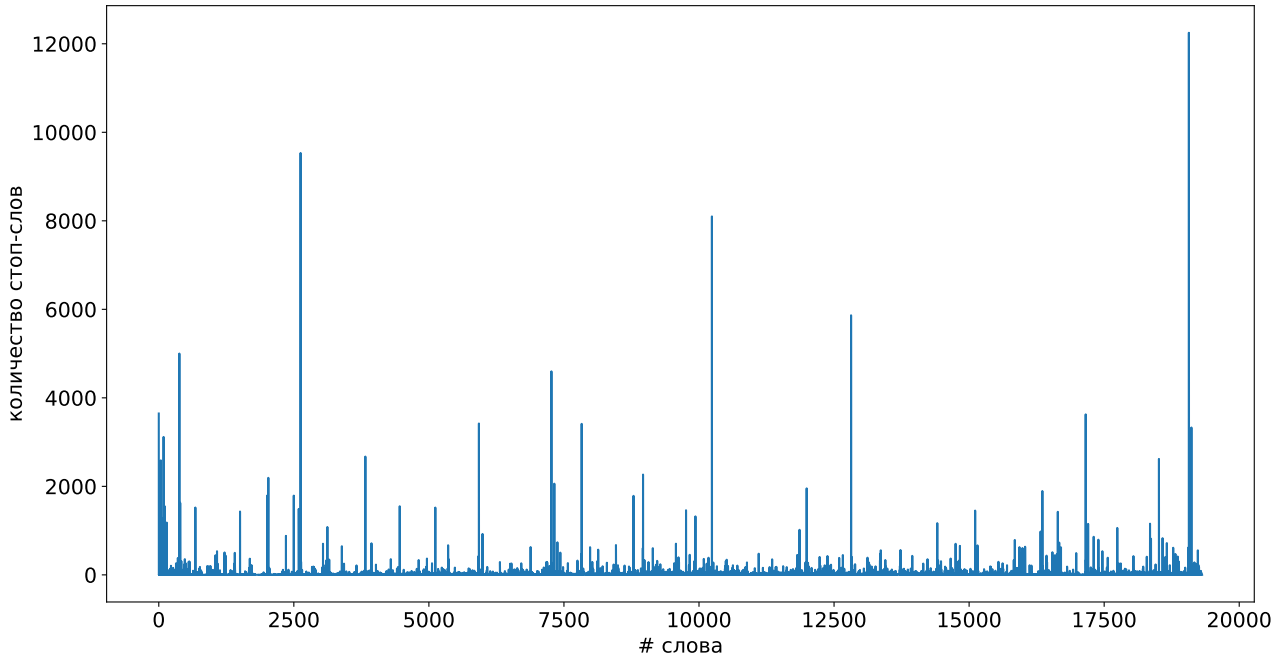


График количества стоп-слов для каждого документа

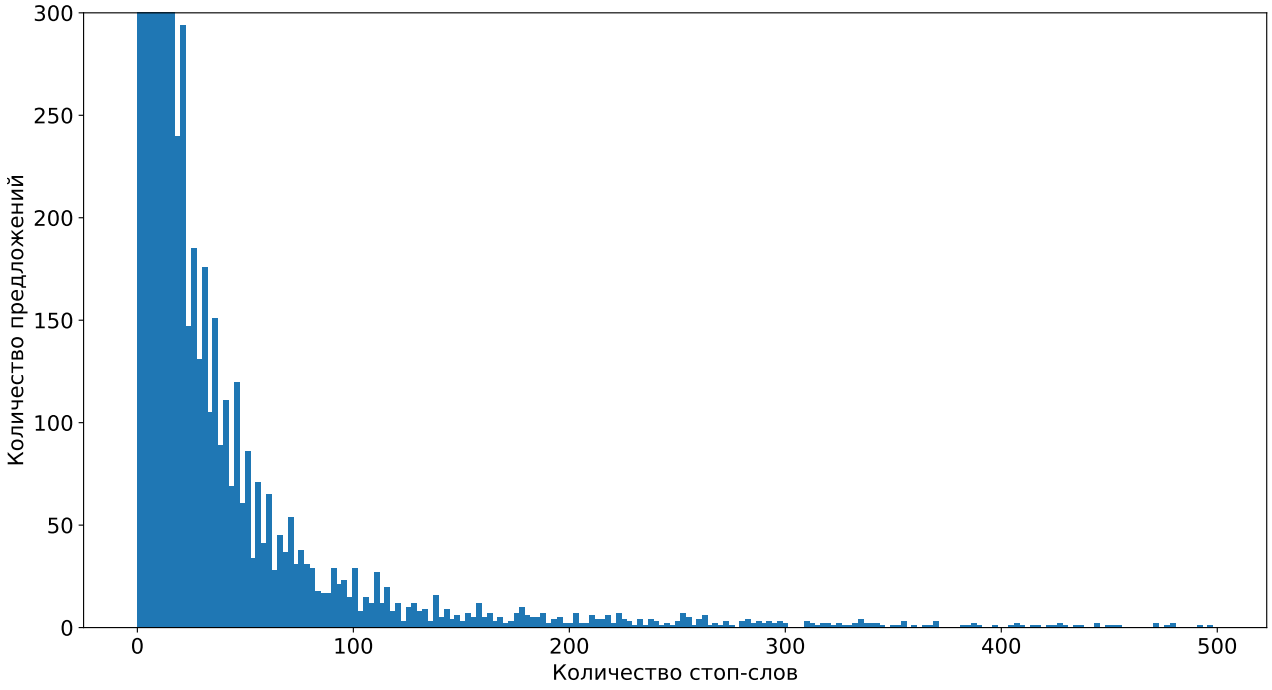


График плотности документов в зависимости от количества стоп-слов (ордината была ограничена 300 словами)

Из графиков видно, что изначальный датасет тексты обработаны, но не удалены стоп-слова.

- Токены были отстемлены. Стемминг использовался только английский, т.к. стемминг очень долго выполняется. Стеммер брался Snowball из `nltk` (`nltk.stem.snowball.EnglishStemmer`)

Лемматизация не использовалась по двум причинам: первая – очень долго работает, вторая – существует только для английского языка.

2.2 Создание признакового пространства

Были созданы четыре признаковых пространства:

- Word2Vec

  Для создания пространства, полученного с помощью Word2Vec была использована библиотека `gensim`. А именно модель `models.Word2Vec` с размером вектора 300. Word2Vec возвращает вектор для слова, поэтому бралось среднее по всем словам из документа. Таким образом пространство для документа – среднее по всем векторам слова документа, полученных с помощью Word2Vec

- Тематическое моделирование

  Для создания пространства, полученного с помощью тематического моделирования была использована библиотека `gensim`. А именно модель `models.LdaModel` с 300 темами. В итоге модель возвращает вероятность принадлежности предложения к каждой теме. Пространство для документа – вероятности принадлежности документа к каждой теме.

- Doc2Vec

  Для создания пространства, полученного с помощью Doc2Vec была использована библиотека `gensim`. А именно модель `models.Doc2Vec` с размером вектора 300. Пространство генерировалось аналогично Word2Vec.

- Tf-idf + SVD

  Для создания пространства, полученного с помощью Tf-idf была использована библиотека `sklearn`. А именно модель `feature_extraction.text.TfidfVectorizer`. Далее матрица tf-idf сжималось с помощью `TruncatedSVD` из `sklearn.decomposition` до размера 300. В итоге получаем пространство из сжатого до 300 признаков tf-idf.

2.3 Классификация

Для классификации использовалась логистическая регрессия. Логистическая регрессия использовалась из `sklearn`. Но так как логистическая регрессия – бинарный классификатор, использовался обертка-классификатор OneVsRest также из `sklearn`. Классификатор использовался с параметрами по-умолчанию.

В итоге получились следующие результаты:

Признаковое пространство	**ROC AUC**	**PR AUC**
Word2Vec	0.97129	0.52899
Тематическое моделирование	0.92817	0.26927
Doc2Vec	0.97101	0.51168
Tf-idf + SVD	0.95058	0.44940

Из таблицы видно, что PR AUC гораздо меньше, чем ROC AUC. Это связано с тем, что Precision-Recall чувствителен к распределению классов. У нас многоклассовая классификация и классы распределены не равномерно и таким образом мы получаем такой маленький PR AUC.

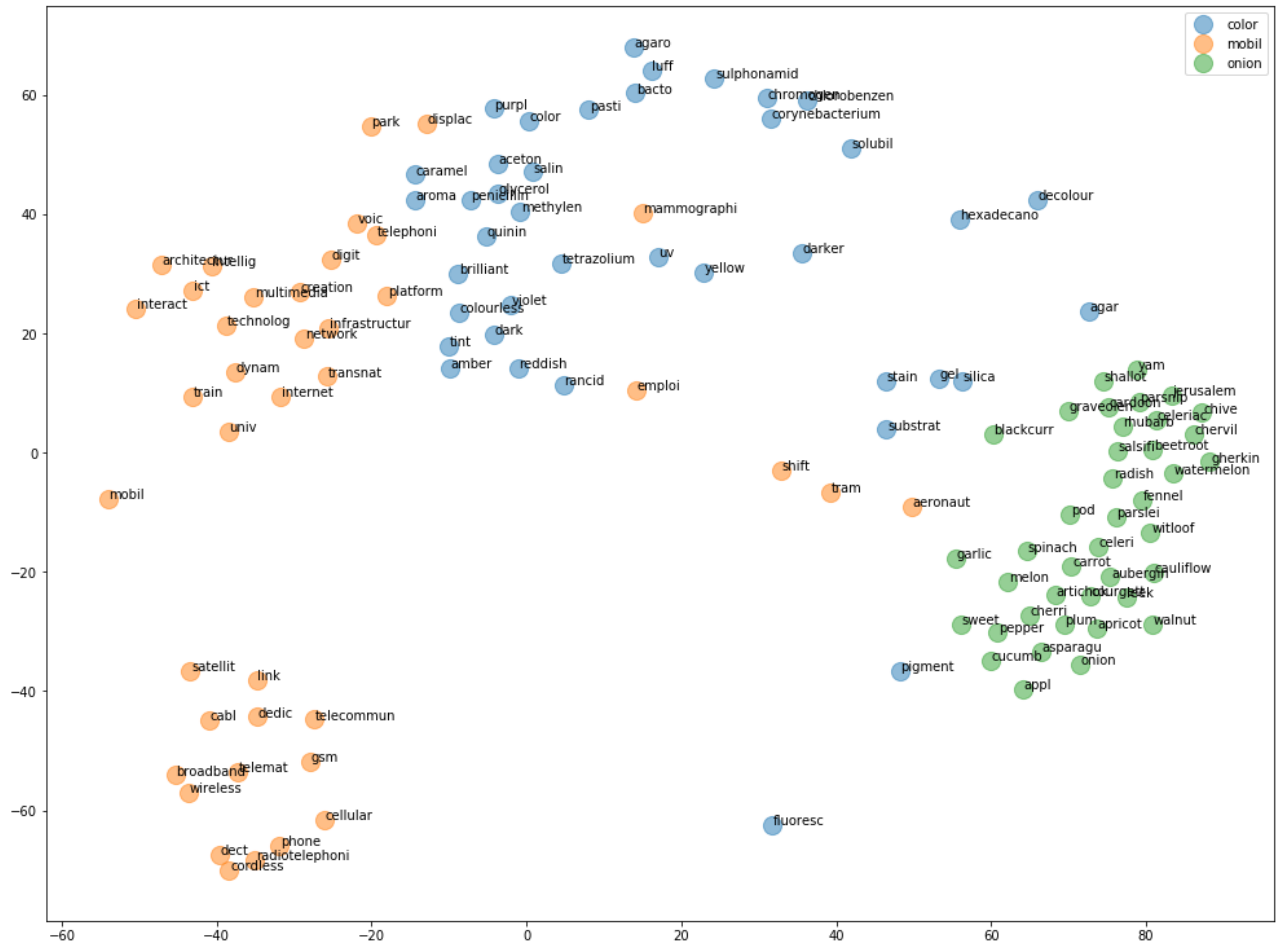
2.4 Сравнение предсказаний

Был реализован метод, который показывает предсказанные метки для документа. Пример данного метода показан в таблице:

Метод	**Классы**
Метки объекта	`ec_agreement, environmental_cooperation, environmental_protection, pollution_control_measures, rhine_valley`
Word2Vec	`ec_cooperation_agreement, united_states, economic_cooperation, ec_agreement, cooperation_agreement`
Тематическое моделирование	`ec_agreement, protocol_to_an_agreement, ec_cooperation_agreement, ec_association_agreement, trade_agreement`
Doc2Vec	`ec_cooperation_agreement, economic_cooperation, united_states, eu_country, cooperation_agreement`
Tf-idf + SVD	`ec_cooperation_agreement, ec_agreement, eu_country, economic_cooperation, cooperation_policy`

3 Визуализация признакового пространста

3.1 Word2Vec

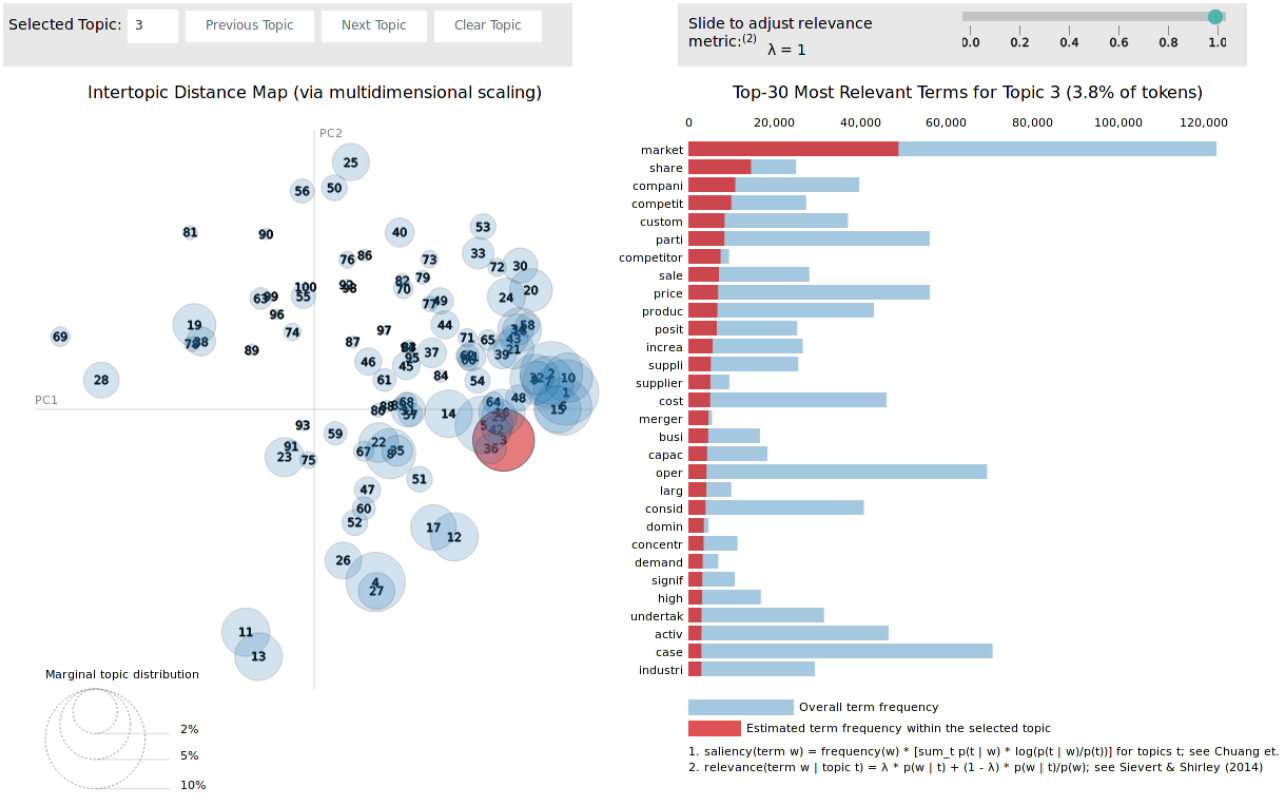


Визуализация для Word2Vec

Для визуализации векторов, полученных с помощью Word2Vec были выбраны 3 слова, значения которых далеки друг от друга: *color, mobil, onion*. Далее брались по 45 близких слов к каждому слову. Для каждого их всех слов считался вектор с помощью Word2Vec размера 100. Затем просторанство этих векторов были уменьшены до пространства размерности 2 с помощью t-SNE преобразования.

В итоге получились связные кластера (т.к. мы брали бликие слова) и эти кластера лежат по отдельности друг от друга (т.к. мы брали 3 слова (элементы каждого кластера) далеких друг от друга). Также, если посмотреть на сами слова, то видно, что они похожи или близки по значению. Таким образом можно сказать, что Word2Vec преобразует слова в вектора таким образом, что он “учитывает” их значения

3.2 LDA



Визуализация LDA

Визуализация тематической модели LDA была сделана с помощью библиотеки `pyLDAvis`. Модель LDA обучалась для 100 тематик, чтобы визуализация была понятной и незагроможденной.

На левом графике визуализации изображено взаимное расположение тематик в проекции на двумерное пространство. При наведении на тематику отрисовывается правый график. На правом графике изображена гистограмма распределний Топ-30 релевантных слов тематики. Красным цветом изображены количества слов в данной тематике, синим – количество данных слов в корпусе.

4 Анализ тональности текстов постов Twitter

4.1 Предобработка данных

Из датасета постов из Twitter нам нужны всего два столбца: Sentiment – класс (0 или 1) и SentimentText – текст документа.

Над данными (SentimentText) была проведена следующая обработка:

- Были оставлены только буквы и цифры. Остальные символы были удалены. Использовалось регулярное выражение “`\W`”
- Были удалены стоп-слова. Стоп-слова использовались встроенные в TfidfVectorizer из `sklearn`

- Токены были отстемлены. Стемминг использовался английский. Стеммер брался PorterStemmer из `gensim` (`parsing.porter.PorterStemmer`)

Лемматизация не использовалась по очень простой причине: в Twitter люди пишут не всегда грамматически верно и много сленга.

4.2 Создание признакового пространства

Были созданы два пространства:

- Tf-idf

  Для создания пространства, полученного с помощью Tf-idf была использована библиотека `sklearn`. А именно модель `feature_extraction.text.TfidfVectorizer`. Матрица не сжималась и использовалась полностью.

- Word count

  Для создания пространства, полученного с помощью word count была использована библиотека `sklearn`. А именно модель `feature_extraction.text.CountVectorizer`. Матрица не сжималась и использовалась полностью.

4.3 Классификация

Для классификации использовались логистическая регрессия (`LogisticRegression`) и наивный байес (`MultinomialNB`). Классификаторы использовались из `sklearn`. Классификаторы использовались с параметрами по-умлочанию.

Наивный байес проверялся на двух признаковых пространствах. Хотя в документации и написано, что для MultinomialNB предпочтительнее целочисленное пространство и он работает на них хорошо, но может работать и на tf-idf, tf-idf показал результаты чуть лучше.

Также я предположил, что если оставлять все символы, то качество классификации улучшится. Предположение подтвердилось, но увеличение точности оказалось всего лишь 0.1%.

В итоге получились следующие результаты:

Классификатор	**Признаковое пространство**	**ROC AUC**	**PR AUC**
Логистическая регрессия	**tf-idf**	0.86034	0.85464
Логистическая регрессия	**tf-idf без фильтрации**	0.86165	0.85791
Наивный байес	**word count**	0.84231	0.83520
Наивный байес	**tf-idf**	0.84359	0.83932
Наивный байес	**tf-idf без фильтрации**	0.84811	0.84520

На таблицы видно, что в случае двуклассовой классификации PR AUC примерно равняется ROC AUC. Это объясняется тем, что у нас классы одинаково распределены.

4.4 Визуализация

Для интереса попробуем нарисовать облако позитивных и негативных слов.

Облако позитивных слов

Облако негативных слов

5 Заключение

В данной работе с помощью методов, изученных в курсе были сделаны задачи классификации текстов и определения тональности текстов. Также:

- были опробованы методы получения векторного представления слов: Word2Vec и Doc2Vec
- был опробован метод LDA тематического моделирования
- были визуализированы результаты работ методов Word2Vec и LDA
- были использованы библиотеки для работы с текстами nltk и gensim
- были использованы разные методы классификации и обработки текстов