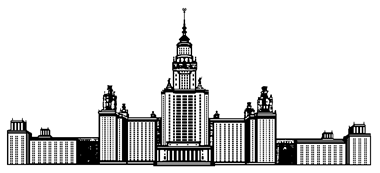
Московский государственный университет имени М.В. Ломоносова

Факультет Вычислительной Математики и Кибернетики

Кафедра Алгоритмических Языков



**ОТЧЁТ СТУДЕНТА 624 ГРУППЫ**

**Выявление факта ссылки на прошедшие события**

Выполнил:

студент 6 курса 624 группы

*Тихомиров Михаил Михайлович*

Москва, 2017 г.

Оглавление

[Описание работы 3](#_Toc469735671)

[Признаки 3](#_Toc469735674)

[Описание программы 3](#_Toc469735675)

[Тестирование 4](#_Toc469735676)

[Анализ полученных результатов и предложения по улучшению 4](#_Toc469735677)

# Описание работы

Работа по извлечению факты ссылки на прошлую информацию в данном решении происходит за счет использования логистической регрессии. Соответственно для решения задачи было необходимо:

1. Сформировать коллекцию
2. Исследовать тематику и придумать информативные признаки
3. Реализовать модуль извлечения признаков
4. Обучить на обучающей коллекции классификатор
5. Протестировать на тестовой коллекции

Коллекция формировалась из новостей с портала lenta.ru

# Признаки

Были выбраны следующие признаки:

1. Количество глаголов в прошедшем времени
2. Номер предложения по отношению к количеству предложений в документе
3. Количество слов в предложении перед первым глаголом в прошедшем времени
4. Наличие слов из списка ключевых слов (Ранее, году, напомним … )

# Описание программы

В рамках данной работы использовался язык python 3. Для разбиения на предложения использовался модуль NTLK. Для получения морфологической информации и разбиения на слова использовался модуль Mystem от Яндекса. Для обучения модели машинного обучения использовалась популярная библиотека scikit learn.

Был выбран Mystem, а не Pymorphy2, так как он лучше справляется со своей задачей.

После обработки одного документа в выходном файле появляются конструкции  
<sentence id=”” learn\_params=”” class=””> … </sentence>

Которые в последующем используются в модуле обучения и тестирования. Атрибут class проставляется позже вручную (разметка коллекции).

На этапе обучения происходит извлечение learn\_params и class для всех предложений обучающей выборки, и по этим векторам происходит обучение модели.

# Тестирование

Были протестированы две модели : svm и логистическая регрессия.

Оценка происходила по двум параметрам:

1. Точность по всем классам
2. Полнота по классу 1, где класс 1 – это класс предложений, которые ссылаются на прошлую информацию.

Результаты показали, что на данных признаках и данной обучающей выборке логистическая регрессия показала себя лучше:

SVM : P = 0.57 R\_1 = 0,33

LogisticRegression : P = 0.77, R\_1 = 0.75

# Анализ полученных результатов и предложения по улучшению

При использовании предложенного метода одним из главных недостатков является то, что объем выборки слишком мал, около 140 объектов. Так же не происходило никакой кросс валидации, поэтому модель могла просто удачно лечь на тестовые примеры.

То, что полнота по классу 1 достигает 75% - весьма хороший результат, но, так как мне важна именно полнота, а не точность, это значение хотелось бы видеть около 90%, даже если точность из-за этого упадет.

Так же было бы полезно изучить влияние отдельных признаков на результат работы программы, что тоже позволит улучшить результаты.

Помимо этого количество ключевых слов очень мало, требуется собрать большую коллекцию, расставить метки и статистически отобрать наиболее частые слова и выражения в предложениях класса 1. Данный пункт может существенно улучшить результаты работы программы.