Московский государственный университет имени
М. В. Ломоносова
Факультет Вычислительной математики и кибернетики
Кафедра Алгоритмических языков

Курсовая работа

Линейные модели в задачах анализа тональности текста

Выполнил:
Студент 324 группы
Давлетов Адис Алмазбекович

Научный руководитель:
к.ф.-м.н.,
Арефьев Николай Викторович

Москва, 2018

Аннотация

В курсовой работе реализуются алгоритмы наивного байесовского классификатора и логистической регрессии. Сравниваются результаты реализованных моделей с известными результатми из [3, 7]. Исследуется влияние гиперпарметров, подходов к обучению на результаты классификации. Исследуются причины, по которым происходят ошибки классификации отзывов. Исследуются признаки, на которые опираются модели, при классификации. Вся работа ведется на датасетах отзывов о фильмах из IMDb [2].

Оглавление

Введение

С фактом того, что скорость появления все новых данных (информации) в глобальной сети растет, спорить не приходится. Ежедневно в различных мессенджерах пользователями генерируются сообщения в больших объемах. На тысячах сайтов оставляются огромное количество комментариев с отзывами о всевозможных продуктах и услугах. И это лишь малая часть гигантского айсберга. Все это ведет к необходимости в инструментах авотматической обработки текстов. Так, у каждого человека ежедневно возникают ситуации, когда необходимо купить какой-то продукт, в чем он не разбирается, или сходить посмотреть фильм в кинотеатре, но хочется чтобы не осталось чувства разочарвания от похода. С учетом того, что количество товаров и услуг и их разнообразие может быть огромно, то, наверняка, была бы полезной система, позволяющая ранжировать все эти товары и услуги. Подобная, хорошо реализованная, система хорошо помогала бы своим пользователям.

Так, одной из задач автоматического анализа текстов является задача построения систем, позволяющих по входному тексту определить его класс (классифицировать): отнести его к одному из заранее определенных классов. В самом простом случае, текст необходимо классифицировать на два класса: например, на положительный и отрицательный классы. Как мы видим, задача построения таких систем, обладающих высокой точностью классификации, является весьма актуальной.

В рамках курсовой работы производится сравнительный анализ существующих моделей, анализ влияния гиперпараметров и подходов к обучению на результат. Ставится целью путем применения метода взвешивания признаков из [5] попробовать улучшить результаты моделей.

Глава 1

Обзор методов обучения

1.1 Наивный байесовский классификатор

Наивный байесовский классификатор — классификатор, основанный на теореме Байеса и «наивном» предположении о том, что события представляющие объект классификации независимы. Объекты классификации в данной модели, как можно было понять из вышесказанного, представляются в виде последовательности некоторых независимых событий. Классификация осуществляется выбором того класса, у которой максимальна условная вероятность события c_i при условии выполнения событий, описывающих указанный объект [1].

$$P(c_i|o) = \frac{P(o|c_i)P(c_i)}{P(o)} = \frac{P(x_1,\ldots,x_n|c_i)P(c_i)}{P(x_1,\ldots,x_n)} = \frac{P(x_1|c_i)\ldots P(x_n|c_i)P(c_i)}{P(x_1)\ldots P(x_n)}$$

$$i = \arg_j \max P(c_j|o)$$

В формулах выше введены следующие обозначения:
o - это объект классификации,
c_i - событие, что объект классификации принадлежит i-му классу,
$x_1,\ldots,x_n$ - независимые события, представляющие объект классификации и
i - результат работы классификатора, класс к которому соотнес классификатор поданый ему объект o.

Так как при классификации нам интересен лишь максимум условных вероятностей $P(c_j|o)$, можно обойтись без вычисления вероятности объекта классификации $P(o)$, т.е. без вычисления вероятностей $P(x_1),\ldots,P(x_2)$

В задачах анализа тональности текста объектом классификации является документ (некоторый текст) и событиями, которые мы «наивно» предполагаем независимыми, являются слова появляющиеся в них.

Подходы к подсчету условных вероятностей слов $P(x_k|c_i)$ и вероятностей классов $P(c_i)$ могут быть различными. Например, можно брать значением $P(x_k|c_i)$ нормированную частоту появления слова x_k в документах данного класса c_i, а вероятность класса $P(c_i)$ - отношением количества документов класса c_i к общему числу документов в обучающей выборке:

$$P(x_k|c_i) = \frac{\#x_k \; in \; c_i}{\#words \; in \; c_i} \quad P(c_i) = \frac{\#documents \; of \; c_i}{\#total \; documents}$$

Согласно [3], результаты обучения и последующей классфикации на данных из IMDb [2] на положительные и отрицательные классы с использованием улучшенной версии этого алгоритма дали точность 88.80%

1.2 Логистическая регрессия

Логистическая регрессия (для бинарной классификации) — статистическая модель, которая по обучающей выборке пытается подобрать параметры модели θ_k так, чтобы гиперплоскость, задаваемая ими, разделяла как можно больше данных из одного, заранее определенного класса c_1 из обучающей выборки от данных из второго класса c_2 [4].

Результат модели лежит в вероятностном пространстве.

$$h(x, \theta) = \frac{1}{1 + \exp(-x^T\theta - \theta_0)}$$

$x = (x_1, \ldots, x_n)$ - вектор признаков объекта классификации,
$\theta = (\theta_1, \ldots, \theta_n)$, θ_0 - параметры модели (веса),
n - количество признаков

Классификатор соотносит объект классификации x к классу c_1, при $h(x, \theta) \geq 0.5$ и ко второму классу c_2, в противном случае.

1.3 NBSVM

Та же самая логистическая регрессия в задачах двухклассовой классификации с определенным образом формируемым вектором признаков для объекта классификации [5]. Объектом классификации является текст (мешок слов). Имеется два класса: положительный и отрицательный. Вектор признаков формируется как логарифм отношения частот слова в положительном и отрицательном классах.

Глава 2

Основная часть

2.1 Постановка задачи

В курсовой работе ставились следующие задачи:

1. Реализовать следующие модели на языке python и сравнить их результаты:

   (a) Наивный байесовский классификатор

   (b) Логистическая регрессия

   Реализовать данные модели для задачи классификации текстов (сентимент-анализа) для двух классов: положительного и отрицательного. Сравнить их результаты.

2. Проанализировать влияние гиперпараметров на результат:
   Поэкспериментировать с моделями, с гиперпараметрами моделей, выяснить какие гиперпараметры влияют на результат сильнее, а какие слабее.

3. Проанализировать ошибки:
   Определить количество неверно классифицированных положительных и отрицательных отзывов.

4. Проанализировать признаки, на которые опираются модели:
   Сравнить признаки, на которые опираются модели логистической регрессии со взвешиванием признаков и без него при решении отнести отзыв к определенному классу.

2.2 Реализованные модели

Во всех моделях производится предобработка текста: все тексты приводятся в нижний регистр, удаляются все небуквенные и нецифровые символы. Удаляются слова из списка стоп-слов из приложения к работе. Строится некоторый словарь на основе предобработанных текстов. В модели наивного байеса к словарю добавляется специальное слово «UNK», которым мы обозначаем все слова не вошедшие в словарь. В остальных моделях такие слова игнорируются. Обучающая выборка включает по 12500 положительных и отрицательных отзывов о фильмах. Тестовая выборка также включает по 12500 примеров для отрицательного и положительного классов. При оптимизации функций ошибок используется оптимизационный метод adagrad [6].

2.2.1 Наивный байесовский классификатор (модель Бернулли)

В модели каждому тексту отзыва сопоставляется бинарный вектор d размерности V, где V – размер построенного словаря. Компоненты d_i вектора d принимают значения 0 и 1:

$$d_i = \begin{cases} 1, \ word\, i\, in\, text \\ 0, \ otherwise \end{cases}$$

Вводятся гиперпараметры $primaryVocabSize$ и $secondaryVocabSize$. Из всех слов из обучающей выборки в словарь берутся $primaryVocabSize$ самых частотных слов. Для каждого слова x_k из словаря подсчитываются его условные вероятности $P(x_k|pos)$ и $P(x_k|neg)$. Проблема нулевой вероятности для незнакомых слов в классе решается **лапласовским сглаживанием**: когда мы считаем, что все слова из словаря появились в классе на один раз больше, чем есть на самом деле.

$$P(x_k|c_i) = \frac{\#documents_{\ni x_k}\ of\ c_i + 1}{\#documents\ of\ c_i + 2} \quad P(c_i) = \frac{\#documents\ of\ c_i}{\#total\ documents}$$

Далее для каждого слова x_k вычисляется его **байесовский вес**, как отношение вероятностей в положительном и отрицательном классах. Строится новый словарь, включающий по $secondaryVocabSize$ слов с самыми большими и с самыми маленькими байесовскими весами. Заново вычисляются все параметры модели, но с уже новым словарем, меньшего размера.

2.2.2 Наивный байесовский классификатор (мультиномиальная модель)

В данной модели также вводятся два гиперпараметра $primaryVocabSize$ и $secondaryVocabSize$ и применяется тот же подход к формированию результирующего словаря. Тексту отзыва здесь сопоставляется уже не бинарный вектор размерности V, а вектор d длиной в число уникальных слов из словаря, встречающихся в тексте, с компонентами d_i:

$$d_i = \#x_i \; in \; text$$

$$d_{UNK} = \begin{cases} 1, \; \exists x_k : x_k \; in \; text, \; but \; not \; in \; vocabulary \\ 0, \; otherwise \end{cases}$$

Все слова, встретившиеся в тексте, но не вошедшие в словарь, в векторе d учитываются единожды как одно слово UNK. Условные вероятности слов $P(x_k|pos)$ и $P(x_k|neg)$ вычисляются следующим образом:

$$P(x_k|c_i) = \frac{\#documents_{\ni x_k} \; of \; c_i + 1}{\#vocabulary \; words \; in \; c_i + |V|} \quad P(c_i) = \frac{\#documents \; of \; c_i}{\#total \; documents}$$

$$P(UNK|c_i) = \frac{\#UNK \; in \; c_i + 1}{\#vocabulary \; words \; in \; c_i + |V|}$$

В обеих моделях, чтобы определить класс текста для каждого класса вычисляется сумма произведений компонент вектора, соответствующего тексту, на соответствующие вероятности слов в этом классе. Классом текста выбирается тот класс, в котором максимальна эта сумма.

2.2.3 Логистическая регрессия

Текст в данной модели представляется как вектор признаков размерности V. Каждая компонента данных векторов представляет некоторое слово из словаря и содержит значение веса данного слова. За вес слова принимается логарифм его **байесовского веса** [5]. При таком подходе представления текстов, мы штрафуем (зануляем) те слова, которые появляются в одинаковой степени как в положительных, так и в отрицательных отзывах. И, наоборот, выделяем те слова, которые встречаются часто в одном классе, но редко в другом.

Размеры	500	1000	3000	6000	8000	10000	10200
20000	84.288	86.992	87.64	82.232	76.004	66.308	–
25000	83.676	86.78	88.224	85.368	80.908	75.024	–
35000	82.236	85.224	87.98	87.716	85.724	82.484	–
60000	77.764	82.752	87.148	88.36	88.392	88.232	–
70000	76.4	82.04	86.924	88.324	88.484	88.656	**88.688**

Таблица 2.1: Точность модели Бернулли при гиперпараметрах primaryVocabSize и secondaryVocabSize на униграммах и биграммах

Размеры	500	1000	2000	2500	2730	3000	6000
12000	86.668	87.324	84.864	81.912	80.312	78.476	52.472
20000	85.008	87	87.496	87.068	86.632	85.94	71.432
24000	84.244	86.788	87.408	87.664	87.464	87.216	77.332
26000	83.924	86.58	87.58	87.652	87.72	87.548	79.3
27000	83.716	86.58	87.508	87.724	87.84	**87.868**	80.204
30000	83.136	86.084	87.464	87.628	87.592	87.628	82.252

Таблица 2.2: Точность мультиномиальной модели при гиперпараметрах primaryVocabSize и secondaryVocabSize на униграммах и биграммах

2.3 Эксперименты

2.3.1 Наивный байесовский классификатор

В моделях наивного байесовского классификатора эксперименты проводятся с включением биграммов и триграммов, так как при использовании только униграммов результаты классификатора получаются поскромнее. Количество уникальных слов (последовательностей слов) при использовании униграммов и биграммов получается более 1.5 миллиона, а при включении еще и триграммов, то более 4 миллионов.

В таблице 2.1 приведены результаты экспериментов с моделью Бернулли. При значениях primaryVocabSize и secondaryVocabSize равных 70000 и 10200 для данной модели был получен результат 88.688.

Результаты экспериментов для мультиномиальной модели представлены в таблице 2.2. Здесь при значениях гиперпараметров 27000 и 3000 удалось достичь точности – 87.868.

В таблицах 2.3 и 2.4 приведены результаты экспериментов при добавлении триграммов. Лучший результат для мультиномиальной модели получился при значениях параметров 70000 и 5000 – 88.652. Для модели Бернулли при значениях 100000 и 10000 – 89.052. Как видим, добавле-

Размеры	1000	3000	5000	7000	10000	20000
50000	84.592	88.22	88.568	88.436	86.188	66.556
70000	82.688	87.884	88.636	88.716	88.624	78.972
100000	80.576	86.528	87.896	88.976	**89.052**	85.692
150000	–	–	86.624	87.676	88.644	89.028
250000	–	–	84.196	85.432	86.72	88.832
500000	–	–	84.176	85.56	86.004	87.56

Таблица 2.3: Точность модели Бернулли при гиперпараметрах primaryVocabSize и secondaryVocabSize на униграммах, биграммах и триграммах

Размеры	1000	3000	5000	7000	10000	20000
50000	84.592	88.22	88.124	85.896	79.824	54.208
70000	82.688	87.884	**88.652**	88.432	85.528	65.784
100000	80.576	86.528	87.992	88.636	88.424	78.848
150000	–	–	86.78	87.492	88.364	86.276
250000	–	–	84.912	85.6	86.428	88.416
500000	–	–	85.012	85.584	85.908	87.116

Таблица 2.4: Точность мультиномиальной модели при гиперпараметрах primaryVocabSize и secondaryVocabSize на униграммах, биграммах и триграммах

ние тирграммов позволило улучшить результаты для мультиномиальной модели с 87.868 до 88.652 и для модели Бернулли с 88.688 до 89.052.

2.3.2 Логистическая регрессия

В ходе эксперимента модель обучалась без использования взвешивания, так и с его использованием. Устанавливается максимальное количество итераций равное 15000, проводится подсчет функции ошибок каждые 100 итераций, устанавливаются коэффициент сходимости равным $1e-05$ и коэффициент скорости обучения равным $4e-02$.

Введем следующие обозначения:
1g – униграммы, 2g – униграммы + биграммы, 3g – униграммы + биграммы + триграммы.

Как видно из рисунков 2.1 и 2.2, использование взвешивания признаков значительно повышает точность классификаторов. На рисунках изображена зависимость функций ошибок на тестовой выборке от количества итераций оптимизационного алгоритма, размера словаря и коэф-

Методы	Результат
MNB + 3g	88.652
BNB + 3g	89.052
LR + 1g	88.172
LR + 1g + взвешивание	88.976
LR + 2g	90.56
LR + 2g + взвешивание	91.412
LR + 3g	90.912
LR + 3g + взвешивание	**91.928**

Таблица 2.5: Результаты экспериментов для различных подходов к обучению

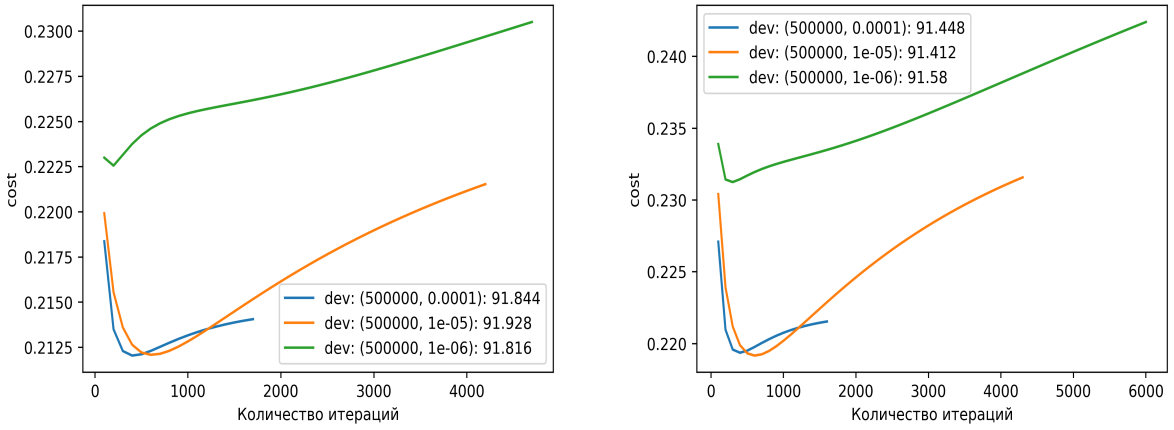

Рис. 2.1: Зависимость функции ошибок от количества итераций в ходе оптимизации параметров модели для 3g (слева) и 2g (справа) с применением взвешивания

фициента регуляризцации. Там же отражены результаты классификаторов после окончания обучения. На рисунках мы можем видеть явление переобучения, когда в процессе обучения функция ошибок на тестовой выборке начиная с некотрой итерации переходит от убывания к возрастанию. В таких случаях мы получаем достаточно высокие результаты для обучающей выборки и низкие (относительно) для тестовой. Такая картина наблюдается для малых значений коэффициента регуляризации. При больших значениях коэффициента регуляризации мы наблюдаем явление недообучения, когда результаты на обеих выборках невысокие.

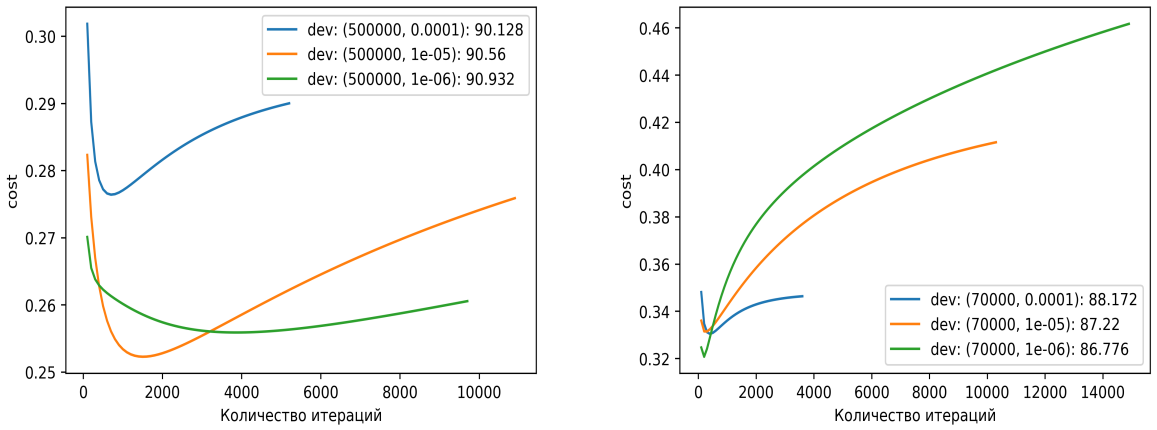

Рис. 2.2: Зависимость функции ошибок от количества итераций в ходе оптимизации параметров модели для 2g (слева) и униграммов (справа) без применения взвешивания

2.3.3 Результаты

В таблице 2.5 приведены результаты проведенных экспериментов. В ходе экспериментов нам удалось улучшить результат указанный в [7] для линейной модели. Также улучшили результат из [3] для наивного байесовского классификатора.

Для задач анализа тональности текста логистическая регрессия работает значительно лучше наивного байесовского классификатора. Как мы видим, отрыв составляет порядка 2%.

Как видно из таблицы, при взвешивании признаков результат увеличивается примерно на процент. При добавлении триграмма рост точности составляет половину процента. При включении еще и четырехграммов, нам удалось получить результат 92.076 для логистической регрессии со взвешиванием признаков.

2.4 Ошибки

Рассматриваются модели логистической регрессии после обучения с N-граммами до триграммов включительно, размером словаря равным 500000 и коэффициентом регуляризации равным $1e-05$ с применением взвешивания и без него.

В модели со взвешиванием признаков было допущено 2018 ошибок, причем количество неправильно классифицированных положительных отзывов составляет 998, а отрицательных 1020.

Отзыв	Увер.	Отв.	Знач.
1. A really realistic, sensible movie by Ramgopal Verma . No stupidity like songs as in other Hindi movies. Class acting by Nana Patekar. Much similarities to real 'encounters'.	-0.088	neg	pos
2. It's not Citizen Kane, but it does deliver. Cleavage, and lots of it. Badly acted and directed, poorly scripted. Who cares? I didn't watch it for the dialog.	-2.25	neg	pos
3. I don't care if some people voted this movie to be bad. If you want the Truth this is a Very Good Movie! It has every thing a movie should have. You really should Get this one.	-0.764	neg	pos
4. The plot was really weak and confused. This is a true Oprah flick. (In Oprah's world, all men are evil and all women are victims.)	0.278	pos	neg
5. Masterpiece. Carrot Top blows the screen away. Never has one movie captured the essence of the human spirit quite like Chairman of the Board. 10/10... don't miss this instant classic.	4.007	pos	neg
6. A dedicated Russian Scientist dreams of going to Mars. He eventually gets there but it takes the whole film before we are able to have a laugh at the Russian style of Revolution in Mars.	0.157	pos	neg

Таблица 2.6: Примеры ошибок при классификации обученной моделью логистической регрессиии со взвешиванием признаков

Во второй модели ошибок всего 2272, неверно классифицированных положительных отзывов 1107, отрицательных же 1165.

Как видно, в обеих моделях допускается примерно одинаковое количество ошибок в положительных и отрицательных отзывах.

Примеры текстов с ошибками классификации с применением взвешивания представленны в таблице 2.6.

2.5 Признаки моделей

В таблице 2.7 приводятся по 20 признаков для моделей логистической регрессии со взвешиванием и без взвешивания, на которые они опираются при классификации. Можно видеть, что есть признаки, сильно

Модель со взвешиванием		Модель без взвешивания	
Положительные признаки	Отрицательные признаки	Положительные признаки	Отрицательные признаки
and does	see why	attention	see why
a trip	4 10	and does	if you have
the extreme	released and	the worst	green
was interesting	dawson	otherwise	couple
true life	the cabin	many other	fun
needed a	the jokes	a trip	all of
save the day	slater	the extreme	stand
was happening	dr hackenstein	was interesting	i like
perception	worst acting	you have a	people are
not appear	profession	bobby	she doesn
beginning of this	everything you	what a	movie was
him how	tara reid	human	the jokes
you have a	shaggy	maybe it	playing
leno	blazing	like this	i never
many other	programs	bad guys	4 10
s set	roman	the american	help
so obvious	burn it	ratings	acting was
only complaint	binder	sound	all of the
criticisms	traffic	was in	but even
acting in	she doesn	edge	killer

Таблица 2.7: Ниболее важные признаки, на которые опираются модели при классификации в порядке убывания

влияющие результат и присутствующие в обеих моделях.

Глава 3

Заключение

В рамках курсовой работы были реализованы модели наивного байесовского классификатора и логистической регрессии для классификации отзывов о фильмах на полжительные и отицательные классы. Экспериментальным путем было выявлено, что логистическая регрессия классифицирует тексты точнее, чем наивный байесовский классификатор. С целью улучшить результаты модели логистической регрессии был применен подход к обучению (предложенный в [5]), который позволил добиться цели. В результате, на N-граммах до триграммов включительно, модель выдала результат 91.928, а при включении еще и четырехграммов, результат получился равным 92.076. Установлено, что применение взвешивания помогает лучше, чем добавление очередного N-грамма. В ходе обучения, в моделях без применения взвешивания и с его применением, формируются разные опорные признаки, сильно влияющие на результат. Есть опорные признаки присутствующие как в одной, так и в другой модели.

Литература

[1] Rish, Irina. (2001). «An empirical study of the naive Bayes classifier» IJCAI 2001 Workshop on Empirical Methods in Artificial Intelligence

[2] Internet Movie Database

[3] Vivek Narayanan, Ishan Arora, Arjun Bhatia
Fast and accurate sentiment classification using an enhanced Naive Bayes model

[4] Andrew Ng, Stanford CS229 Lecture Notes

[5] Sida Wang and Christopher D. Manning
Baselines and Bigrams: Simple, Good Sentiment and Topic Classification

[6] John Duchi, Elad Hazan, Yoram Singer
Adaptive Subgradient Methods for Online Learning and Stochastic Optimization

[7] Gregoire Mesnil, Tomas Mikolov, Marc'Aurelio Ranzato, Yoshua Bengio
Ensemble of generative and discriminative techniques for sentiment analysis of movie reviews

[8] Mohit Iyyer, Varun Manjunatha, Jordan Boyd-Graber, Hal Daume III
Deep Unordered Composition Rivals Syntactic Methods for Text Classification

Приложение А

Стоп-слова

['i', 'me', 'my', 'myself', 'we', 'our', 'ours', 'ourselves', 'you', 'your', 'yours', 'yourself', 'yourselves', 'he', 'him', 'his', 'himself', 'she', 'her', 'hers', 'herself', 'it', 'its', 'itself', 'they', 'them', 'their', 'theirs', 'themselves', 'what', 'which', 'who', 'whom', 'this', 'that', 'these', 'those', 'am', 'is', 'are', 'was', 'were', 'be', 'been', 'being', 'have', 'has', 'had', 'having', 'do', 'does', 'did', 'doing', 'a', 'an', 'the', 'and', 'but', 'if', 'or', 'because', 'as', 'until', 'while', 'of', 'at', 'by', 'for', 'with', 'about', 'against', 'between', 'into', 'through', 'during', 'before', 'after', 'above', 'below', 'to', 'from', 'up', 'down', 'in', 'out', 'on', 'off', 'over', 'under', 'again', 'further', 'then', 'once', 'here', 'there', 'when', 'where', 'why', 'how', 'all', 'any', 'both', 'each', 'few', 'more', 'most', 'other', 'some', 'such', 'no', 'nor', 'not', 'only', 'own', 'same', 'so', 'than', 'too', 'very', 's', 't', 'can', 'will', 'just', 'I', 'should', 'now', 'd', 'll', 'm', 'o', 're', 've', 'y', 'ain', 'aren', 'couldn', 'didn', 'doesn', 'hadn', 'hasn', 'haven', 'isn', 'ma', 'mightn', 'mustn', 'needn', 'shan', 'shouldn', 'wasn', 'weren', 'won', 'wouldn']