Классификация документов с помощью векторных представлений

Шундеев Александр Сергеевич
кандидат физико-математических наук
Название учебного заведения или научной организации; или должности и места работы
МГУ имени М.В. Ломоносова
119192 Москва, Мичуринский проспект, дом 1
alex.shundeev@gmail.com

Балахничев Сергей Александрович
студент
МГУ имени М.В. Ломоносова
serg.balah@gmail.com

Заславский Давид Дмитриевич
студент
МГУ имени М.В. Ломоносова
zabaf@ya.ru

Пехтерев Станислав Игоревич
студент
МГУ имени М.В. Ломоносова
stas-19000@mail.ru

Аннотация: Статья содержит обзор современных моделей векторных представлений слов. Подобные модели рассматриваются в контексте решения задачи классификации документов. Исследуются свойства, как исходных векторных представлений слов, так и векторных представлений слов, подвергшихся пост-обработке.

Ключевые слова: векторное представление слов, векторное представление документов, классификация текстов, Word2Vec, GloVe, fastText, сингулярное разложение, латентный семантический анализ, дистрибутивная гипотеза.

Document classification using word embeddings

Alexandr S. Shundeev
Candidate of Physical and Mathematical Sciences
Lomonosov Moscow State University
Moscow, ul. Michurinskiy prospekt, d. 1, Russia 119192
alex.shundeev@gmail.com

Sergey A. Balakhnichev
Student
Lomonosov Moscow State University
serg.balah@gmail.com

David D. Zaslavskii
Student
Lomonosov Moscow State University
zabaf@ya.ru

Stanislav I. Pekhterev
Student
Lomonosov Moscow State University
stas-19000@mail.ru

Abstract: The article contains an overview of modern models of word embeddings. The models are considered in the context of the document classification problem. The properties of both the original word embeddings and the word embeddings with post-processing are investigated.

Keywords: word embeddings, document embeddings, text classification, Word2Vec, GloVe, fastText, singular value decomposition, latent semantic analysis, distributional hypothesis.

Введение

В настоящее время сложилась ситуация, при которой методы интеллектуального анализа данных все чаще становятся основой для построения прикладных информационных систем. Можно с большой долей уверенности спрогнозировать, что в будущем подобная тенденция не только сохранится, но и усилится. В результате корректность постановок задач в области анализа данных, а также правильный выбор и реализация методов для их решения, будут критическим образом влиять на успешность разработки, сопровождения и эксплуатации прикладных информационных систем.

Современным и бурно развивающимся подходом в области анализа текстовых данных является использование так называемых векторных представлений слов (word embeddings), которые выступают основным объектом исследования в рамках данной работы. Векторное представление слов представляет собой соответствие между словами и вещественными векторами фиксированной размерности. При построении векторного представления слов пытаются достигнуть следующей цели. Близким по смыслу словам должны соответствовать близкие вектора. Подобное построение осуществляется в рамках некоторой модели, имеющей ряд настраиваемых параметров, и основывается на обработке входного корпуса текстов. О процессе построения векторного представления слов говорят также как о процессе обучения.

Проиллюстрируем введенные понятия на примере. В открытом доступе имеется целый ряд построенных наборов векторных представлений слов, которые можно использовать в образовательных и исследовательских целях. Как правило, подобные наборы предоставляются авторами соответствующих моделей векторных представлений слов и призваны продемонстрировать их преимущества. В частности, создатели модели GloVe (Global Vectors for Word Representation) [1] подготовили наборы[1] векторных представлений слов размерностей 50, 100, 200 и 300, которые были получены в результате обработки текстов статей англоязычной Википедии 2014 года. На рис. 1 изображены проекции нескольких 50-мерных векторов из этих наборов на двухмерную плоскость. Каждая проекция (точка) подписана соответствующим вектору словом. Невооруженным глазом видно, что близкие по смыслу слова сгруппированы рядом друг с другом.

Следует более формально определить, как при работе с векторными представлениями слов определяется близость между вещественными векторами, и что понимается под смысловой близостью слов.

На практике используются разные подходы для определения близости между векторами, в том числе косинусная близость, евклидово расстояние, метрика Манхэттена, расстояние Бхаттачарья, расстояние Хеллингера, дивергенция КульбакаЛейблера. В работе [2] на примере решения ряда задач показывается, что наилучшие результаты достигаются с использованием косинусной близости (косинус угла между векторами). Значение косинусной близости равное единицы трактуется как максимальное сходство между векторами и соответствующими им словами, а нулевое значение трактуется как максимальное различие. Так, например, для слова *red* наиболее близкими оказываются слова *yellow* (0.899), *blue* (0.890), *green* (0.856), *black* (0.840), *purple* (0.832). В скобках указано значение косинусной

[1] https://nlp.stanford.edu/projects/glove

близости. Для слова *cat* наиболее близкими оказываются слова *dog* (0.921), *rabbit* (0.848), *monkey* (0.804), *rat* (0.789), *cats* (0.786).

В компьютерной лингвистике считается, что два слова являются семантически близкими (semantically similar) [3], если они имеют общую родительскую категорию (гипероним, «сверх-имя»). Так *собака* и *кролик* оба являются *животным*. *Москва*, *Берлин*, *Вена* являются столицами. Более общим понятием является семантическая связность (semantic relatedness) слов [4]. Семантическая связность включает в себя такие отношения между словами, как синонимия (например, слова *смелый* и *храбрый*), антонимия (например, слова *выигрыш* и *проигрыш*), меронимия (отношение части и целого), гипонимия (родо-видовое отношение). К семантически связанным словам также относят слова, которые совместно встречаются в рассматриваемом корпусе текстов.

Рисунок 1 – Разбиение набора слов на смысловые группы.

Как было продемонстрировано выше, векторные представления слов могут успешно использоваться для решения задачи определения смысловой близости между словами. При этом дается числовая оценка смысловой близости двух слов. Кроме того, векторные представления слов могут успешно использоваться для изучения смысловых отношений между словами. Например, пары слов (*Москва*, *Россия*), (*Берлин*, *Германия*) и (*Минск*, *Белоруссия*) являются примерами отношения «столица - страна». Подобные пары слов называются аналогиями [5]. В работах [6], [7] была поставлена задача поиска аналогий, имеющая следующую формулировку. Для заданных слов A, B, C необходимо подобрать слово D таким образом, чтобы пары слов (A, B) и (C, D) являлись аналогиями по отношению друг к другу. Например, для слов *Москва*, *Россия*, *Берлин* оптимальным решением будет слово *Германия*.

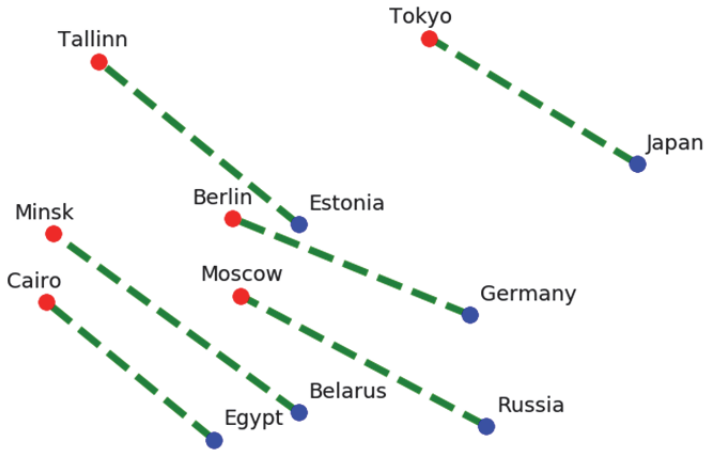

Рисунок 2 – Примеры отношений слов в задаче поиска аналогий.

В ходе поиска подходов для решения этой задачи были разработаны две модели векторных представлений слов под общим названием Word2Vec [6], [8]. Идея найденного решения состоит в следующем. Предположим, что словам A, B, C, D сопоставлены соответственно вектора $\mathbf{v}_A$, $\mathbf{v}_B$, $\mathbf{v}_C$, $\mathbf{v}_D$. Оказалось, что если пары слов (A, B) и (C, D) являются аналогиями, то имеет место приближенное равенство векторов $\mathbf{v}_B - \mathbf{v}_A \approx \mathbf{v}_D - \mathbf{v}_C$ (рис. 2). Поэтому в качестве неизвестного слова D нужно подобрать слово, вектор которого наиболее близок вектору $\mathbf{v}_B - \mathbf{v}_A + \mathbf{v}_C$. В рас-

сматриваемом примере задача поиска аналогий для отношения «столица - страна» правильно решается с вероятностью 0.67. При этом вероятность случайного угадывания составляет $0.25 \cdot 10^{-5}$, что может быть признано хорошим результатом.

В настоящее время точность решения задачи определения смысловой близости слов и задачи поиска аналогий является основным критерием качества векторного представления слов. Подобное свойство векторных представлений слов можно использовать и для решения других задач в области обработки текстов на естественном языке.

В данной работе векторные представления слов рассматриваются с позиций решения задачи классификации документов. Они рассматриваются как основа для построения математических моделей документов, к которым применимы методы машинного обучения. Исследуются свойства как исходных векторных представлений слов, полученных в рамках стандартных моделей, так и векторные представления слов, подвергшиеся операции постобработки.

Дальнейшее изложение структурировано следующим образом. В разделе 1 приводится обзор популярных моделей векторных представлений слов и описываются подходы по их построению. В разделе 2 описываются используемые в дальнейшем математические модели документов, и дается формальная постановка задачи классификации документов. В разделе 3 приводятся результаты проведенных экспериментов, на основе которых делаются выводы о целесообразности использования векторных представлений слов в решении задачи классификации документов.

1 Модели векторных представлений слов

Модели векторных представлений слов активно изучаются уже несколько десятилетий. В качестве одной из первых работ на этом направлении можно отметить работу 1975 года [9]. На начальном этапе подобные модели базировались на построении и преобразовании частотных матриц типа «слово - документ» или «слово - контекст». К этому периоду в частности относится создание метода латентного семантического анализа LSA (Latent Semantic Analysis) [10], в рамках которого впервые было обосновано применение сингулярного разложения (Singular Value Decomposition) частотной матрицы для получения векторного представления слов. Частотные (count based) модели подробно описаны в обзорной работе [11].

В последнее время наибольшее внимание уделяется так называемым предсказательным (predictive) моделям, в рамках которых векторное представление слов получается как результат решения некоторой оптимизационной задачи. Интерес к предсказательным моделям можно связать с общим ростом популярности нейросетевых методов и подходов. Так в 2003 году была разработана вероятностная нейросетевая модель NPLM (Neural Probabilistic Language Model) [12]. На протяжении следующих десяти лет эта модель постепенно упрощалась. Результатом таких упрощений стало появление семейства моделей Word2Vec [6], [8], которые представляют собой нейронные сети с одним скрытым слоем, не содержащие нелинейных преобразований.

Преимуществом предсказательных моделей является возможность обучаться на большом объеме исходных текстовых данных. Ответить однозначно в рамках, каких (частотных или предсказательных) моделей можно получить более качественные векторные представления слов, не представляется возможным. На этот вопрос существуют диаметрально противоположные ответы [13], [14].

Прежде, чем перейти к рассмотрению конкретных моделей векторных представлений слов сформулируем гипотезы из области компьютерной лингвистики, которые положены в основу их построения. Первая гипотеза, получившая название дистрибутивной (distributional hypothesis) [15], [16], [17], утверждает, что слова, появляющиеся в похожих контекстах внутри корпуса текстов, скорее всего, будут иметь похожий смысл.

В качестве контекста может выступать весь документ или его отдельные фрагменты (предложение, абзац, глава в книге). Часто, контекст задается окном некоторого фиксированного размера. Окном является последовательность слов в документе, отстоящих не далее, чем на h позиций от заданного слова w. Само слово w в окно не входит. Контекстом может быть любое слово в окне, множество или мультимножество всех слов окна.

Частотные модели также базируются на гипотезе мешка слов (bag of words hypothesis) [9], что смысл документа не зависит от порядка слов, которые в нем встречаются.

1.1 Частотные матрицы

В основе частотных моделей лежит понятие частотной матрицы. Выделяют три типа таких матриц [11]. Предположим, что задан корпус документов, и зафиксирован словарь, состоящий из всех слов, встречающихся в этих документах. В матрице типа «слово - документ» (word – document) $\boldsymbol{X} = (x_{wd})$ строки соответствуют словам, а столбцы соответствуют документам. Элемент матрицы x_{wd} равен числу вхождений слова w в документ d.

Таблица 1. Пример исходной и взвешенной (TF-IDF) матрицы типа «слово - документ».

w \ d	исходная			взвешенная		
	d_1	d_2	d_3	d_1	d_2	d_3
i	1	1	0	0.4	0.4	0
love	1	0	0	1.09	0	0
dogs	1	1	0	0.4	0.4	0
hate	0	1	0	0	1.09	0
and	0	1	1	0	0.4	0.4
hunting	0	1	1	0	0.4	0.4
is	0	0	1	0	0	1.09
my	0	0	2	0	0	2.18
hobby	0	0	1	0	0	1.09
passion	0	0	1	0	0	1.09

В таблице 1 приведен пример матрицы типа «слово - документ», составленной по следующему набору документов: d_1 «*i love dogs*», d_2 «*i hate dogs and hunting*», d_3 «*hunting is my hobby and my passion*».

Матрицы типа «слово - документ» является частным случаем матриц тип «слово - контекст» (word - context). В этом случае вместо документов рассматриваются всевозможные контексты. Если в качестве контекстов выступают отдельные слова, то говорят о матрицах типа «слово - слово» (word - word). Матрица $\boldsymbol{X} = (x_{wc})$ этого типа является квадратной и симметричной. О ней также говорят как о матрице счетчиков совместной встречаемости слов.

Таблица 2. Пример частотной матрицы типа «слово - слово».

w \ c	*the*	*quick*	*brown*	*fox*	*jumps*	*over*	*lazy*	*dog*
the	0	1	1	2	1	1	1	1
quick	1	0	1	1	1	0	0	0
brown	1	1	0	1	1	1	0	0
fox	2	1	1	0	1	1	0	0
jumps	1	1	1	1	0	1	1	0
over	1	0	1	1	1	0	1	1
lazy	1	0	0	0	1	1	0	1
dog	1	0	0	0	0	1	1	0

В таблице 2 представлена матрица типа «слово - слово», составленная для предложения «*the quick brown fox jumps over the lazy dog*». Контекст задается окном размера $h = 3$.

Описанные выше матрицы, как правило, обладают несбалансированностью. Значения элементов строки, соответствующей часто встречающемуся слово, будут во много раз больше значений элементов строки, соответствующих редкому слову. Поэтому обычно частотная матрица подвергается преобразованию «взвешивания» ее элементов. В случае матриц типа «слово - документ» могут применяться функции семейства TF-IDF [18]. Например, в качестве нового значения элемента матрицы можно взять величину $x_{wd} = tf(w,d) \times idf(w)$, где $tf(w,d)$ – частота слова (term frequency) w в документе d, а $idf(w)$ - обратная частота документа (inverse document frequency). Обратная частота документа является логарифмом отношения общего числа документов в рассматриваемом корпусе к числу документов, в которых слово w встречается хотя бы один раз.

В случае матриц типа «слово - слово» старое значение ее элемента x_{wc} может быть заменено на значение функции

$$PMI(w,c) = \log\frac{p(w,c)}{p(w)p(c)},$$

которая называется поточечной взаимной информацией (Pointwise Mutual Information) [19]. Величина $p(w,c)$ является эмпирической вероятностью появления слова w в контексте c. Величины $p(w)$ и $p(c)$ являются соответственно эмпирическими вероятностями появления слов w и c в рассматриваемом корпусе документов. В работе [20] предлагается использовать положительную поточечную взаимную информацию (Positive Pointwise Mutual Information)

$$PPMI(w,c) = \max\big(0, PMI(w,c)\big).$$

В ней обосновывается, что среди всех подходов к взвешиванию частотных матриц применение этой функции дает наилучшие результаты при решении задачи определения смысловой близости.

1.2 Частотные модели на основе сингулярного разложения

Частотная матрица (исходная или взвешенная) задает векторное представление слов. Действительно, каждому слову по определению соответствует вектор-строка в такой матрице. Однако подобное векторное представление слов обладает двумя недостатками. Первым недостатком является большая размерность таких векторов, совпадающая с числом всех документов в рассматриваемом корпусе или с размером словаря. Большая размерность фактически исключает возможность практического использования такого векторного представления слов ввиду неприемлемого объема сопутствующих вычислительных затрат и высокой вычислительной погрешности. Второй недостаток состоит в том, что вычисленное расстояние между такими векторами плохо отражает меру смысловой близости между соответствующими этим векторам словами.

Тем не менее, существует подход, позволяющий на основе частотной матрицы построить малоразмерное векторное представление слов, адекватно описывающее смысловую близость между словами. Этот подход базируется на использовании метода сингулярного разложения матриц [21]. Произвольная вещественная матрица $\boldsymbol{X}$ может быть представлена в виде

$$\boldsymbol{X} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T.$$

В этом разложении матрицы $\boldsymbol{U}$ и $\boldsymbol{V}$ имеют ортонормированные столбцы ($\boldsymbol{U}\boldsymbol{U}^T = \boldsymbol{I}$, $\boldsymbol{V}\boldsymbol{V}^T = \boldsymbol{I}$, $\boldsymbol{I}$ – единичная матрица). Матрица $\boldsymbol{\Sigma}$ представляет собой диагональную матрицу сингулярных значений и имеет одинаковый с матрицей $\boldsymbol{X}$ ранг r.

Пусть выбрано число $\mathrm{k} < r$. Через Σ_k обозначим диагональную матрицу, составленную из k верхних сингулярных значений. Соответственно, через $\mathrm{U_k}$ и $\mathrm{V_k}$ обозначим матрицы, составленные из столбцов матриц U и V, соответствующих k верхним сингулярным значениям. Имеет место следующее приближение исходной матрицы

$$\boldsymbol{X} \approx \widehat{\boldsymbol{X}} = \boldsymbol{U}_k\boldsymbol{\Sigma}_k{\boldsymbol{V}_k}^T.$$

Приближение $\widehat{\mathrm{X}}$ является наилучшим в том смысле, что на нем достигается минимум величины $\left\|\widehat{\mathrm{X}} - \mathrm{X}\right\|_\mathrm{F}$ среди всех матриц ранга k. Здесь $\|...\|_\mathrm{F}$ обозначает норму Фробениуса.

Строки матрицы $\mathrm{U_k}$ могут интерпретироваться как векторное представление слов малой размерности k. Применение описанной процедуры к частотной матрице типа «слово - документ», взвешенной с помощью функции TF-IDF составляет основу метода латентного семантического анализа [10].

1.3 Модели Word2Vec и fastText

Под общим названием Word2Vec скрываются две модели векторных представлений слов [6], [8]: CBOW (continuous bag-of-words) и SG (skip-gram). В обеих моделях контекст слова задается окном некоторого фиксированного размера h. Разберем эвристики, которые положены в основу этих моделей.

В качестве примера рассмотрим следующее предложение:

the quick brown fox jumps over the lazy dog.

Предположим, что $h = 3$. Тогда окно контекста для слова *fox* будет состоять из последовательности слов *the*, *quick*, *brown*, *jumps*, *over*, *the*, как это показано ниже:

the quick brown *fox* ***jumps over the*** *lazy dog.*

Модель CBOW пытается предсказать, какое слово может появиться в заданном контексте. Например, в следующем предложении:

the quick brown *?* ***jumps over the*** *lazy dog,*

на пропущенном месте с большой долей вероятности должно быть восстановлено слово *fox*.

Модель SG решает обратную задачу, пытаясь по заданному слову, восстановить контекст, в котором оно могло бы появиться:

? *?* *?* *fox* *?* *?* *?* *dog.*

Модели Word2Vec имеют два настраиваемых параметра: размер окна контекста h и размерность целевого векторного представления слов n. В качестве входных данных выступает корпус документов, использующих некоторый словарь D. В ходе построения векторного представления слов для каждого слова $w \in D$ вычисляются два вектора $\mathbf{v}_w, \acute{\mathbf{v}}_w \in \mathbb{R}^n$. Итоговое векторное представление слов образуют вектора $\mathbf{v}_w$.

Дальнейшее описание приведем для случая модели CBOW и $h = 1$. Условная вероятность появления слова w в контексте слова c моделируется с помощью выражения вида

$$p(w|c) = \frac{\exp\langle \acute{\mathbf{v}}_w, \mathbf{v}_c\rangle}{\sum_{s\in D}\exp\langle \acute{\mathbf{v}}_s, \mathbf{v}_c\rangle}.$$

На основе подобных условных вероятностей записывается функция правдоподобия, для которой решается задача максимизации

$$L(\Theta) = \prod_{(w,c)\in T} p\,(w|c;\Theta) \longrightarrow \max_\Theta,$$

где $\Theta = \{(\mathbf{v}_w, \acute{\mathbf{v}}_w)|w \in D\}$, а T – множество всевозможных пар (слово, контекст), составленных на основе входного корпуса документов. Итоговое векторное представление слов является частью решения этой оптимизационной задачи.

Дальнейшее, развитие моделей Word2Vec пошло по двум направлениям. В рамках первого направления были разработаны модели векторных представлений документов Doc2Vec [22]. На основе модели CBOW была построена модель DM (distributed memory), а на основе модели SG была построена модель DBOW (distributed bag-of-words). Основная идея моделей Doc2Vec состоит в расширении словаря. Каждому документу ставится в соответствие уникальный ключ. Эти ключи рассматриваются как псевдослова и добавляются в словарь. Считается, что ключ документа встречается в любом контексте этого документа. Соответственно, в рамках модели DM по контексту предсказывается ключ документа, а в рамках модели DBOW по ключу документа предсказывается контекст. Векторное представление документов составлено из векторов, вычисленных для их ключей.

В рамках второго направления была обобщена модель SG, на основе которой была разработана модель векторного представления слов fastText [23], ориентированная на работу с морфологически сложными языками. Предполагается, что слова в рассматриваемом корпусе документов записаны в алфавите, не содержащем символы < и >. Далее, каждому слову в исходном словаре слева приписывается символ <, а справа приписывается символ >. Например, слово *where* будет преобразовано в слово <*where*>.

Модель fastText имеет дополнительный натуральный параметр *N*. Для заданного слова *N*-граммой называется любая последовательность символов длины *N*, встречающаяся в этом слове. Например, 3-граммами слова <*where*> будут последовательности символов <*wh*, *whe*, *her*, *ere* и *re*>. Для всевозможных *N*-грамм, встречающихся в словах модифицированного словаря, строится векторное представление. Для этого решается задача предсказания контекста по *N*-грамме слова, встречающегося в этом контексте. В итоговом векторном представлении каждому слову ставится в соответствие сумма векторов всех *N*-грамм, соответствующих этому слову.

1.4 Модель GloVe

Настраиваемыми параметрами модели GloVe [1] являются размерность векторного представления слов n и размер контекста h. По входному корпусу документов строится частотная матрица типа «слово - слово» $\boldsymbol{X} = (x_{wc})$. На основе частотной матрицы $\boldsymbol{X}$ можно вычислить условную вероятность появления слова $w \in D$ в контексте слова $c \in D$ вида

$$p(w|c) = \frac{x_{wc}}{\sum_s x_{ws}}.$$

Авторы делают следующее эвристическое предположение. Пусть заданы три слова $w, c_1, c_2 \in D$. Исследуя отношение условных вероятностей

$$\frac{p(w|c_1)}{p(w|c_2)}$$

можно сделать определенные выводы о семантической связанности этих трех слов друг с другом. Таблица 3 иллюстрирует эти закономерности. В ней в качестве c_1 выбрано слово *ice*, а в качестве c_2 выбрано слово *steam*. В качестве w последовательно выступают слова *solid*, *gas*, *water*, *fashion*. Так, например, если слово w (*solid*) семантически связано с контекстом c_1 и не связано с контекстом c_2, то отношение условных вероятностей будет сравнительно большим. Если наоборот, слово w (*gas*) семантически связано с контекстом c_2 и не связано с контекстом c_1, то отношение будет мало. Если одновременно слово w (*gas, fashion*) семантически связано или не связано с контекстами c_1 и c_2, то отношение условных вероятностей будет приблизительно равно единице.

Таблица 3. Иллюстрация эвристики, заложенной в основу модели GloVe (взято из [1]).

Статистика \ w	$solid$	gas	$water$	$fashion$
$p(w\|ice)$	1.9×10^{-4}	6.6×10^{-5}	3.0×10^{-3}	1.7×10^{-5}
$p(w\|steam)$	2.2×10^{-5}	7.8×10^{-4}	2.2×10^{-3}	1.8×10^{-5}
$\frac{p(w\|ice)}{p(w\|steam)}$	8.9	8.5×10^{-2}	1.36	0.96

На основе приведенных соображений авторами модели ставится следующая оптимизационная задача, в рамках решения которой строится векторное представление слов:

$$J(\Theta) = \sum_{w,c \in D} f(x_{wc})\left(\langle \mathbf{v}_w, \acute{\mathbf{v}}_c \rangle + b_w + \acute{b}_w - \log x_{wc}\right)^2 \longrightarrow \min_{\Theta},$$

где $\Theta = \left\{\left(\mathbf{v}_w, \acute{\mathbf{v}}_w, b_w, \acute{b}_w\right) \middle| w \in D\right\}$ ($\mathbf{v}_w, \acute{\mathbf{v}}_w \in \mathbb{R}^n$, $b_w, \acute{b}_w \in \mathbb{R}$). В качестве векторных представлений слов может использоваться каждый из следующих трех наборов векторов $\{\mathbf{v}_w\}$, $\{\acute{\mathbf{v}}_w\}$ или $\{\mathbf{v}_w + \acute{\mathbf{v}}_w\}$. В то же время авторы модели рекомендуют использовать третий набор, использование которого в ходе проведенных экспериментов показывает наилучшие результаты при решении задачи определения смысловой близости слов и задачи поиска аналогий.

В определении функционала J фигурирует весовая функция f. Эта функция является непрерывной и монотонно неубывающей. В нуле она принимает нулевое значение. С помощью этой функции штрафуются слишком большие значения счетчиков x_{wc}. В качестве весовой функции авторы модели рекомендуют использовать функцию вида

$$f(x) = \begin{cases} (x \,/\, x_{max})^{3/4}, & x < x_{max}; \\ 1, & \text{иначе.} \end{cases}$$

2 Задача классификации документов

В теории машинного обучения дается формальная постановка задачи классификации. В то же время следует отметить, что методы машинного обучения применимы только к математическим объектам, которыми тексты на естественном языке не являются. Поэтому, чтобы иметь возможность использовать понятия и методы машинного обучения применительно к задаче классификации документов, необходимо вначале заменить документы на их представления в рамках некоторой математической модели.

2.1 Математические модели текстов

В частотной матрице типа «слово – документ» каждому документу из рассматриваемого корпуса соответствует свой столбец. Такой столбец можно рассматривать как векторное представление документа. К подобным векторным представлениям можно применять методы машинного обучения для решения задачи классификации документов. Такое векторное представление можно интерпретировать как математическую модель документов.

Исходная частотная матрица, к которой не применялась процедура взвешивания элементов, порождает модель документов под названием «мешок слов» BoW (Bag of Words). Частотная матрица, к элементам которой была применена функция взвешивания из семейства TF-IDF [18], порождает TF-IDF модель документов.

Если задано некоторое векторное представление слов, то на его основе можно построить векторное представление документов. Наиболее распространенный подход заключается в суммировании векторов всех слов, встречающихся в документе. После этого полученная сумма усредняется (делится на количество слов в документе). Если некоторое слово встречается в документе несколько раз, то при суммировании и усреднении учитывается каждое его появление в документе.

2.2 Машинное обучение

В общем виде задача классификации имеет следующую постановку. Должны быть заданы множество объектов $\mathcal{X}$, и конечное множество классов $\mathcal{Y}$. В дальнейшем, в качестве объектов будут выступать документы, точнее их представления в соответствующей математической модели. Предполагается, что существует неизвестная функциональная зависимость между объектами и классами, о которой можно судить только по конечному множеству обучающих примеров $T = \{(x_i, y_i) | i = 1, \dots, m\} \subset \mathcal{X} \times \mathcal{Y}$.

Решение задачи классификации осуществляется в рамках некоторой модели обучения $M = (H, a)$. Подобная модель включает в себя множество гипотез H (функций вида $h: \mathcal{X} \to \mathcal{Y}$), среди которых ищется приближение к неизвестной функциональной зависимости, а также алгоритм a. Этот алгоритм для множества обучающих примеров T выбирает гипотезу $a(T) \in H$, которая трактуется как решение задачи классификации. Выбранную гипотезу называют классификатором, а выбор гипотезы интерпретируют как процесс обучения, в рамках которого строится классификатор.

Для оценки качества построенного классификатора используются различные числовые метрики. В общем случае подобная метрика имеет вид $\text{est}(T, h)$ и отражает соответствие гипотезы $h \in H$ множеству обучающих примеров T. Наиболее распространенной метрикой является точность (accuracy)

$$\text{acc}(T, h) = \frac{1}{m} \sum_{i=1}^{m} 1\{h(x_i) = y_i\}.$$

Если гипотеза h правильно классифицирует все объекты из обучающих примеров, то значением метрики acc является 1. Если все объекты из обучающих примеров были классифицированы неправильно, метрика acc принимает значение 0. Несмотря на свою простоту и интуитивную понятность данная метрика обладает существенным недостатком. Этот недостаток может проявляться в ситуации, когда объекты неравномерно распределены по классам. В этой ситуации классификатор, оцененный как высокоточный, может неправильно классифицировать все объекты из некоторых классов.

Этого недостатка лишена метрика F_1, которая изначально применяется в случае бинарной классификации $\mathcal{Y} = \{0,1\}$. В своем определении она использует две вспомогательных метрики

$$F_1(T, h) = 2 \cdot \frac{\text{precision}(T, h) \cdot \text{recall}(T, h)}{\text{precision}(T, h) + \text{recall}(T, h)}.$$

Выделим три группы обучающих примеров. Первая группа $TP(T, h) = \{(x, 1) \in T | h(x) = 1\}$. Вторая группа $FP(T, h) = \{(x, 0) \in T | h(x) = 1\}$. Третья группа $FN(T, h) = \{(x, 1) \in T | h(x) = 1\}$. Тогда

$$\text{precision}(T, h) = \frac{|TP(T, h)|}{|TP(T, h)| + |FP(T, h)|}$$

и

$$\text{recall}(T,h) = \frac{|TP(T,h)|}{|TP(T,h)| + |FN(T,h)|}.$$

Метрика F_1 может быть обобщена на случай $|Y| > 2$. С помощью анализируемой гипотезы h для каждого класса можно рассматривать отдельную бинарную задачу классификации (объект принадлежит этому классу или принадлежит любому другому классу). Следовательно, для каждого класса может быть получена своя F_1 оценка. Полученные оценки можно усреднить, или можно взять их взвешенную сумму.

Важным этапом решения задачи классификации является выбор подходящей модели обучения из ряда альтернатив, либо если модель обучения имеет настраиваемые параметры, требуется для таких параметров подобрать оптимальные значения. Возможны комбинации обозначенных вариантов. Проблема выбора модели тесно связана с необходимостью борьбы с двумя негативными явлениями, возникающими в процессе обучения, которые тесно связаны между собой. Первое явление носит название недообучения (underfitting). Оно возникает в ситуации, когда оценка $\text{est}\big(T, a(T)\big)$ признается неудовлетворительной. Второе явление носит название переобучения (overfitting). Оно диагностируется в ситуации, когда построенный классификатор показывает хорошие результаты только на объектах из обучающих примеров. Наличие явления переобучения говорит о том, что одной оценки $\text{est}\big(T, a(T)\big)$ недостаточно, чтобы судить о качестве классификатора.

Множество обучающих примеров можно разбить на два непересекающихся множества $T = T_{\text{train}} \cup T_{\text{test}}$, называемых соответственно тренировочной и тестовой выборками. Тренировочная выборка, как правило, содержит 70% обучающих примеров. Только примеры из обучающей выборки используются для построения классификатора. Оценка $\text{est}\big(T_{\text{train}}, a(T_{\text{train}})\big)$ показывает, имело ли место недообучение. Сравнивая между собой две оценки $\text{est}\big(T_{\text{train}}, a(T_{\text{train}})\big)$ и $\text{est}\big(T_{\text{test}}, a(T_{\text{train}})\big)$, можно сделать вывод о наличии переобучения.

3 Эксперименты

Настоящий раздел посвящен обсуждению результатов проведенных экспериментов над тестовыми наборами данных. На основе этих результатов делаются выводы о возможности использования векторных представлений слов в решении задачи классификации документов.

3.1 Тестовые наборы данных

В ходе проведения экспериментов было использовано три набора документов: movies (рецензии к кинофильмам), R8 (финансовые документы), twitter (сообщения из одноименной социальной сети). Опишем характеристики каждого из этих наборов.

Набор movies состоит из 44012 документов, разбитых на 6 классов. Вектор (0.477, 0.28, 0.084, 0.079, 0.0451, 0.0266) описывает распределение документов по классам. Словарь состоит из 72295 слов. Максимальный, минимальный и средний размер документа соответственно 698, 5 и 56 слов.

Набор R8 состоит из 7674 документов, разбитых 8 классов. Вектор (0.51, 0.29, 0.048, 0.042, 0.038, 0.035, 0.018, 0.006) описывает распределение документов по классам. Словарь состоит из 17387 слов. Максимальный, минимальный и средний размер документа соответственно 533, 4 и 64.5 слова.

Набор twitter состоит из 1594557 документов, разбитых на 2 класса. Вектор (0.5, 0.5) описывает распределение документов по классам. Словарь состоит из 35738 слов. Максимальный, минимальный и средний размер документа соответственно 50, 1 и 12.75 слова.

Как можно видеть из всех наборов тестовых данных сбалансированным является только набор twitter.

3.2 Построение векторных представлений

В ходе проведения экспериментов была использована модель векторных представлений слов GloVe. В рамках этой модели на основе тестовых наборов документов были построены векторные представления слов, имеющие размерность 50, 100, 300. Назовем эти векторные представления слов исходными. После этого исходные векторные представления слов были подвергнуты дополнительной обработке (пост-обработка).

В области анализа данных существует ряд методов понижения размерности данных [24], представленных в виде вещественных векторов фиксированной размерности. Одним из них является метод главных компонент PCA (Principal Component Analysis). Для большинства наборов данных можно построить прямую (ось, направление), обладающую следующим свойством. Проекции векторов из рассматриваемого набора данных на эту прямую будут порождать максимальное рассеивание. Полученную прямую называют первой главной компонентой. Далее, описанную процедуру можно применить к подпространству, ортогональному первой главной компоненте. В результате будет получена вторая главная компонента и так далее. Обычно считается, что проекции на последние главные компоненты можно безболезненно удалить из данных. При этом информационное наполнение в этих данных не ухудшится, зато серьезно понизятся накладные расходы по их хранению. В ряде работ, в том числе [25], пропагандируется подход, в рамках которого применительно к векторным представлениям слов необходимо удалять проекции не на последние, а на первые главные компоненты.

Для каждого исходного векторного представления слов были вычислены главные компоненты. Переход к новому базису, образованному главными компонентами, порождает новое векторное представление слов. Будем говорить, что новое векторное представление слов имеет тип PCA. Будем говорить, что удаление из нового векторного представления слов первой (последней) координаты порождает векторное представление слов типа PCA-1 (PCA-n). Обратим внимание на то, что удаление первой (последней) координаты соответствует удалению проекции на первое (последнее) главное направление.

На основе исходных векторных представлений слов, а также векторных представлений слов типа PCA, PCA-1, PCA-n были построены векторные представления документов для тестовых наборов. Дополнительно для всех документов из тестовых наборов были построены представления в модели BoW.

3.3 Результаты

В ходе проведения экспериментов были использованы две модели классификации: логистическая регрессия (простая модель) и случайный лес (сложная ансамблевая модель) [24]. Для этих целей были взяты реализации этих моделей из библиотеки Scikit-Learn[2]: класс LogisticRegression, реализующий модель логистической регрессии, и класс ExtraTreesClassifier, реализующий модель случайного леса. Каждая из этих классов имеет набор настраиваемых параметров.

В случае класса LogisticRegression задавался алгоритм решения соответствующей оптимизационной задачи (значения newton-cg, lbfgs, liblinear, sag, saga), параметр регуляризации (значения 1000, 100, 10, 1, 0.1) и начальное значение датчика псевдослучайных чисел (два значения). В случае класса ExtraTreesClassifier задавалось количество деревьев (значения 50, 100, 200), максимальная глубина деревьев (значения 10, 20, 50, 100), минимальный размер выборки, которая может быть подвергнута разбиению, (значения 2, 5. 10) и начальное значение датчика псевдослучайных чисел (два значения). Для каждой комбинации значений настраиваемых параметров и набора входных данных обучался и оценивался отдельный классификатор.

Таблица 4. Результаты классификации, соответствующие модели документов BoW.

Набор	Метрика	Логистическая регрессия	Случайный лес
movies	accuracy	0.878890	0.675916
	F_1	0.815951	0.306914
R8	accuracy	0.974874	0.936957
	F_1	0.940006	0.809663
twitter	accuracy	0.795760	0.782860
	F_1	0.795760	0.782860

В таблице 4 приведены результаты экспериментов над классификаторами, построенными на основе элементарной модели документов BoW. По каждому набору тестовых данных и каждой модели классификации был построен, обучен и оценен целый набор классификаторов. Каждой допустимой комбинации значений настраиваемых параметров модели обучения соответствует свой построенный классификатор. Для каждого классификатора были вычислены значения метрик accuracy и F_1. Максимальные значения этих метрик, достигнутые на тестовой выборке, представлены в таблице 4.

Аналогично устроены таблицы 5 и 6, в которых приводятся результаты экспериментов над классификаторами, построенных на основе векторных представлений слов. Отличие состоит только в том, что в каждой ячейке указано три числовых значения, соответствующих размерностям 50, 100 и 300 использованных векторных представлений слов.

Сравнивая между собой результаты из таблиц 4 и 5, можно сделать следующий вывод. Никаких принципиальных улучшений за счет использования векторных представлений слов в характеристиках построенных классификаторов достигнуто не было. Возможно, причина кроется в том, что была выбрана простейшая модель представления документа, использующая векторные представления входящих в него слов. Возможно, переход к более сложной модели (например, [26]) покажет принципиально другие результаты.

Следует также обратить внимание на то, что использование простейшей модели линейного классификатора дало лучшие результаты, по сравнению со сложной моделью случайного леса. Скорее всего, это следствие переобученности классификаторов, построенных на основе модели случайного леса. Проиллюстрируем это на примере набора movies. На обучающей выборке максимальными значениями метрики accuracy для логистической регрессии будут числа 0.754849, 0.768380, 0.792187. Для случайного леса это будут числа 0.999898, 0.999898, 0.999898. На обучающей выборке максимальными значениями метрики F_1 для логистической регрессии будут числа 0.607060, 0.641528, 0.689451. Для случайного леса это будут числа 0.999953, 0.999953, 0.999953.

В случае логистической регрессии результаты для исходных векторных представлений слов и векторных представлений слов типа PCA оказались практически одинаковыми, что выглядит вполне естественно. Переход от од-

[2] https://scikit-learn.org

ного базиса к другому и соответствующее этому переходу преобразование координат векторов не должно было повлиять на построение и работу линейного классификатора.

В таблице 6 приведены результаты экспериментов для векторных представлений слов типа PCA-1. Они практически совпадают с результатами для векторных представлений слов типа PCA-n, а также исходных векторных представлений слов (таблица 5).

Таблица 5. Результаты классификации, соответствующие исходному векторному представлению слов.

Набор	Метрика	Логистическая регрессия	Случайный лес
movies	accuracy	0.762393 0.771344 0.788075	0.752134 0.735954 0.708620
	F_1	0.612906 0.640898 0.674389	0.554547 0.504728 0.429052
R8	accuracy	0.966195 0.968936 0.972133	0.962540 0.959799 0.962083
	F_1	0.917470 0.924408 0.937034	0.866952 0.874149 0.866870
twitter	accuracy	0.742439 0.755875 0.766018	0.755431 0.756635 0.754625
	F_1	0.742420 0.755861 0.766013	0.755391 0.756597 0.754605

Исходя из этого результата, можно сделать вывод об оправданности применения постобработки векторных представлений в случае решения задачи классификации документов. Качество результатов классификации заметно не ухудшается. В то же время накладные расходы на хранение векторных представлений слов могут быть существенно уменьшены. Остается правда открытым вопрос, какие из главных направлений следует выбирать для удаления.

Таблица 6. Результаты классификации, соответствующие векторному представлению слов типа PCA-1.

Набор	Метрика	Логистическая регрессия	Случайный лес
movies	accuracy	0.759777 0.770724 0.788763	0.744492 0.727072 0.687001
	F_1	0.608684 0.638769 0.674617	0.535313 0.476396 0.359658
R8	accuracy	0.965738 0.967108 0.970763	0.961626 0.955231 0.915943
	F_1	0.913092 0.914673 0.927439	0.869057 0.847996 0.717559
twitter	accuracy	0.742053 0.755725 0.765969	-
	F_1	0.742039 0.755712 0.765964	-

Список использованной литературы

[1] Pennington J., Socher R., Manning C.D. GloVe: Global Vectors for Word Representation, Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2014, pp. 1532-1544.

[2] Bullinaria J.A., Levy J.P. Extracting Semantic Representations from Word Co-occurrence Statistics: A Computational Study, 2007 Behavior Research Methods, vol. 39, pp. 510–526.

[3] Resnik P. Using Information Content to Evaluate Semantic Similarity in a Taxonomy, Proceedings of the International Joint Conference for Artificial Intelligence (IJCAI-95). 1995, pp. 448–453.

[4] Budanitsky A., Hirst G. Semantic distance in WordNet: An experimental, application-oriented evaluation of five measures, Workshop on WordNet and other lexical resources, NAACL, 2001.

[5] Gentner D. Structure-mapping: A theoretical framework for analogy, Cognitive Science, 1983, vol. 7, no. 2. pp. 155–170.

[6] Mikolov T., Sutskever I., Chen K., Corrado G., Dean J. Distributed representations of words and phrases and their compositionality, Proceedings of the 26th International Conference on Neural Information Processing Systems, 2013, vol. 2, pp. 3111-3119.

[7] Mikolov T., Yih W., Zweig G. Linguistic Regularities in Continuous SpaceWord Representations, HLT-NAACL, 2013, pp. 746–751.

[8] Mikolov T., Chen K., Corrado G., Dean J. Efficient Estimation of Word Representations in Vector, Computing Research Repository (CoRR), 2013, pp. 1-12, available at: https://arxiv.org/abs/1301.3781.

[9] Salton G., Wong A., Yang C.S. A Vector Space Model for Automatic Indexing, Commun. ACM, 1975, vol. 18, no. 11, pp. 613–620.

[10] Deerwester S.C., Dumais S.T., Landauer T.K., Furnas G.W., Harshman R.A. Indexing by latent semantic analysis. Journal of the American Society for Information Science (JASIS), 1990, vol. 41, no. 6, pp. 391–407.

[11] Turney P.D., Pantel P. From Frequency to Meaning: Vector Space Models of Semantics, Journal of Artificial Intelligence Research, 2010, vol. 37, pp. 141-188.

[12] Bengio Y., Ducharme R., Vincent P., Janvin C. A Neural Probabilistic Language Model, Journal of Machine Learning Research, 2003, Vol. 3, pp. 1137–1155.

[13] Baroni M., Kruszewski G. Don't count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors, Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, vol. 1, pp. 238–247.

[14] Levy O., Goldberg Y., Dagan I. Improving Distributional Similarity with Lessons Learned from Word Embeddings, TACL, 2015, vol. 3, pp. 211–225.

[15] Wittgenstein L. Philosophical Investigations. Blackwell. Translated by G.E.M. Anscombe, 1953.

[16] Harris Z. Distributional structure, Word, 1954, vol. 10, no. 23, pp. 146-162.

[17] Firth, J. R. (). A synopsis of linguistic theory 1930–1955. In Studies in Linguistic Analysis, Blackwell, Oxford, 1957, pp. 1–32.

[18] Sparck J.K. A statistical interpretation of term specificity and its application in retrieval. Journal of Documentation, 1972, vol. 28, no. 1, pp. 11–21.

[19] Church K., Hanks P. (1989). Word association norms, mutual information, and lexicography. In Proceedings of the 27th Annual Conference of the Association of Computational Linguistics, 1989, pp. 76–83.

[20] Niwa, Y., Nitta, Y. Co-occurrence vectors from corpora vs. distance vectors from dictionaries. In Proceedings of the 15th International Conference On Computational Linguistics, 1994, pp. 304–309.

[21] Golub G.H., Van Loan, C.F. Matrix Computations (Third edition). Johns Hopkins University Press, Baltimore, MD, 1996.

[22] Le Q., Mikolov T. Distributed Representations of Sentences and Documents, Proceedings of the 31st International Conference on Machine Learning, 2014, vol. 32, no. 2, pp. 1188-1196.

[23] Bojanowski P., Grave E., Joulin A., Mikolov T. Enriching Word Vectors with Subword Information, Computing Research Repository (CoRR), 2017, pp. 1-12, available at: https://arxiv.org/abs/1607.04606.

[24] Bishop C.M. Pattern Recognition and Machine Learning, Springer, Science+Business Media LLC, 2006, 738 p.

[25] Mu J., Bhat S., Viswanath P. All-but-the-Top: Simple and Effective Post-processing for Word Representations, Computing Research Repository (CoRR), 2018, pp. 1-25, available at: https://arxiv.org/abs/1702.01417.

[26] Wu L., Yen I.E.H., Xu K., Xu F., Balakrishnan A., Chen P., Ravikumar P., Witbrock M.J. Word Mover's Embedding: From Word2Vec to Document Embedding. Computing Research Repository (CoRR), 2018, pp. 1-15, available at: https://arxiv.org/abs/1811.01713.