Key stages of text processing and feature generation in text classification

Skorokhodov I.[1], Tikhomirova A.[2]

Основные этапы обработки текста и генерации признаков в задачах текстовой классификации

Скороходов И. С.[1], Тихомирова А. Н.[2]

[1]Скороходов Иван Сергеевич / Skorokhodov Ivan – бакалавр менеджмента, магистрант,
кафедра экономики и менеджмента в промышленности,
факультет управления и экономики высоких технологий;
[2]Тихомирова Анна Николаевна / Tikhomirova Anna – кандидат технических наук, доцент,
кафедра кибернетики, факультет кибернетики,
Федеральное государственное автономное образовательное учреждение высшего образования,
Национальный исследовательский ядерный университет,
Московский инженерно-физический институт, г. Москва

Аннотация: *в данной работе исследуются основные этапы проведения обработки текстовых признаков в задачах интеллектуального анализа данных, а также процедуры генерации информативных факторов. Среди них рассматриваются операции стемминга, лемматизации, формирования мешка слов, формирования статистики TF-IDF, а также методы сокращения пространства признаков.*
Abstract: *in this paper we investigate key stages of text features processing, which are typically used in data mining tasks, as well as describe main procedures of generating informative factors. Among them we depict such operations as stemming, lemmatization, bag of words and TF-IDF metrics generation and methods to reduce feature space.*

Ключевые слова: *интеллектуальный анализ данных, текстовые признаки, компьютерная лингвистика, обработка данных, генерация признаков.*
Keywords: *data mining, text features, computational linguistics, data processing, feature generation.*

Подготовка данных тесно связана с моделированием и оценкой качества модели, так как в процессе каждого из последующих этапов может обнаружиться информация, которая может быть использована для повышения точности предсказаний [1].

Ключевая цель подготовки данных и генерации признаков — это преобразовать свойства объектов таким образом, чтобы алгоритм смог понять различия между ними и увидеть закономерность, которая порождает их распределение [2].

В общем случае обработка сырых текстовых данных состоит из трех последовательных этапов: устранение грамматических и лексических ошибок, лемматизация и стемминг.

Устранение грамматических и лексических ошибок — это крайне сложная процедура, изучением которой занимается целое направление компьютерной лингвистики [3]. Алгоритм работы должен знать не только правила нужного языка, но и множество исключений. Разработка программного обеспечения требует квалифицированной команды лингвистов и специалистов по машинному обучению. Поэтому на сегодняшний день самый качественный способ провести подобную процедуру для небольших исследовательских задач — это воспользоваться продуктами крупных компаний.

Одна из самых сложных задач в проектах машинного обучения, связанных с обработкой естественного языка, это понимание семантики слов, точнее, генерация признаков таким образом, чтобы алгоритм имел возможность различать понятия, а не наборы букв.

Стемминг — это процесс нахождения основы слова для заданного исходного слова. Основа слова необязательно совпадает с его морфологическим корнем [4].

Для обработки текста на английском языке самыми популярными стеммерами являются стеммер Портера, Snowball-стеммер и стеммер Ланкастера [5]. Snowball-стеммер является улучшением мягкого стеммера Портера. Стеммер Ланкастера — наиболее агрессивный среди перечисленных стеммеров, но, благодаря этому, он более производителен. Лемматизация (в компьютерной лингвистике) — это процесс определения леммы слова [6], то есть канонической, основной его формы. Лемматизация является более сложной процедурой, чем стемминг, так как выявление леммы слова должно основываться на контексте. Например, слово «meeting» может быть как глаголом, так и существительным. Также лемматизация оставляет больше различий между словами. Сравнение стемминга с лемматизацией на примере предложения «A cat in gloves catches no mice» представлено в таблице 1.

Таблица 1. Пример стемминга и лемматизации

Изначальное слово	Слово после лемматизации	Слово после обработки Snowball-

		стеммером
A	a	A
cat	cat	Cat
in	in	in
gloves	glove	glove
catches	catch	catch
no	no	no
mice	mice	mouse

Как видно из примера в таблице 1, лемматизация и стемминг достаточно по-разному ведут себя на одних и тех же словах. В некоторых задачах отличия стемминга от преимущества могут быть преимуществом, а в некоторых — наоборот. Поэтому зачастую используют оба подхода для того, чтобы проверить на практике, какой из них сработает лучше [7].

Извлечение признаков — это процесс построения информативных признаков из исходных, которые в будущем приведут к более быстрому обучению или могут лучше интерпретироваться [8].

Генерация признаков — это процесс и процедура создания и извлечения числовых признаков из сырых данных, которые можно подать на вход какой-либо модели для обучения.

Качественные признаки должны простым образом отражать «закон природы», который обеспечивает их распределение. В задачах, связанных с обработкой естественного языка, существует набор стандартных практик для генерации признаков. К ним относят удаление шумовых слов, создание мешка слов и использование TF-IDF.

Шумовые слова (или стоп-слова) — термин из теории поиска информации по ключевым словам [9]. Это такие слова, знаки, символы, которые самостоятельно не несут никакой смысловой нагрузки, но которые, тем не менее, совершенно необходимы для нормального восприятия текста, его целостности.

Шумовые слова могут делиться на общие и зависимые. К общим можно отнести предлоги, суффиксы, причастия, междометия, цифры, частицы и т. п.

Общие шумовые слова всегда исключаются из поискового запроса (за исключением поиска по строгому соответствию поисковой фразы). Считается, что каждое из общих стоп-слов есть почти во всех документах коллекции.

Зависимые шумовые слова зависят относительно поисковой фразы. Идея заключается в том, чтобы по-разному учитывать отсутствие просто слов из запроса и зависимых стоп-слов из запроса в найденном документе.

Шумовые слова не несут в себе практически никакой смысловой нагрузки, следовательно, не являются важным подпространством для нахождения закономерности. Поэтому удаление шумовых слов обеспечивает более быструю сходимость [10].

В задачах, связанных с обработкой естественного языка, основными признаками являются словесные. Основной способ формирования признаков из слов — это представление всех слов в виде так называемого «мешка». Суть мешка слов заключается в кодировании всех слов выборки в единый словарь и создания пространства дихотомических или порядковых переменных, каждое измерение в котором отражает количество раз, какое слово с данным индексом встретилось в документе:

$$D = D_1 \cup D_2 \cup \ldots \cup D_n, \quad (1)$$

где D_i — это множество слов в объекте i, n — количество объектов.

Размерность признакового пространства при этом становится равной количеству уникальных нешумовых слов во всей выборке, а матрица признаков становится сильно разреженной. Обработка большого количества признаков является вычислительно очень трудоемкой задачей, поэтому перед обучениям применяются различные методы понижения размерности. Также разреженность признаков стоит учитывать при выборе численных методов нахождения оптимального значения функционала качества.

Во многих задачах значение имеет то, насколько часто те или иные слова встречаются в различных документах. Это обуславливается тем фактом, что именно редко встречаемые слова и характеризуют объект — общеупотребимые слова обычно служат в качестве «обвязки» речи. Для того чтобы добавить вес редким терминам и понизить веса общих слов, используют метрику TF-IDF. TF-IDF — (от англ. TF — term frequency, IDF — inverse document frequency) — статистическая мера, используемая для оценки важности слова в контексте документа, являющегося частью коллекции документов или корпуса. Вес некоторого слова пропорционален количеству употребления этого слова в документе и обратно пропорционален частоте употребления слова в других документах коллекции.

TF (term frequency — частота слова) — отношение числа вхождения некоторого слова к общему количеству слов документа. Таким образом, оценивается важность слова t_i в пределах отдельного документа.

$$TF(t,d) = \frac{n_i}{\sum^k n_k}, \quad (2)$$

где:

- n_i — это число вхождений слова в документ,
- n_k — общее число слов в данном документе.

IDF (inverse document frequency — обратная частота документа) — инверсия частоты, с которой некоторое слово встречается в документах коллекции. Учёт IDF уменьшает вес широкоупотребительных слов. Для каждого уникального слова в пределах конкретной коллекции документов существует только одно значение IDF.

$$IDF(t,D) = log\frac{|D|}{|(d_i \supset t_i)|} \quad (3)$$

где:

- $|D|$ — количество документов в корпусе;
- $|(d_i \supset t_i)|$ — количество документов, в которых встречается t_i (когда $n_i \neq 0$).

Выбор основания логарифма в формуле не имеет значения, поскольку изменение основания приводит к изменению веса каждого слова на постоянный множитель, что не влияет на соотношение весов.

Таким образом, мера TF-IDF является произведением двух сомножителей:

$$TF - IDF(t,d,D) = tf(t,d) \times idf(t,D). \quad (4)$$

Большой вес в TF-IDF получат слова с высокой частотой в пределах конкретного документа и с низкой частотой употреблений в других документах. Таким образом, TF-IDF является улучшением мешка слов.

Подобное преобразование текста в числовые признаки позволяет алгоритму воспринимать сходство и различие между объектами и искать зависимости в данных на их основе.

Матрица признаков, получаемая после создания мешка слов, получается крайне большой: размерность пространства равна количеству уникальных слов в имеющемся корпусе:

$$n = \sum_{j=1}^{k} |\bigcup_{i=1}^{\ell} \mu_i^j|, \quad (5)$$

где:

- n — размерность итогового пространства признаков;
- k — количество исходных текстовых описаний объекта;
- ℓ — количество объектов в выборке;
- μ_i^j — j-ое текстовое свойство i-ого объекта.

С учетом того, что матрица сильно разрежена, ее использование в исходном виде становится еще более нецелесообразным. Поэтому прибегают к латентно-семантическому анализу, который позволяет на основе выявляемых между текстами и словами взаимосвязей отбирать только самые важные признаки.

Латентно-семантический анализ можно сравнить с простым видом нейросети, состоящей из трех слоев: первый слой содержит множество слов, второй — некое множество документов, соответствующих определенным ситуациям, а третий, средний - скрытый слой представляет собой множество узлов с различными весовыми коэффициентами, связывающих первый и второй слои.

Наиболее распространенный вариант латентно-семантического анализа основан на использовании разложения диагональной матрицы по сингулярным значениям. С помощью сингулярного разложения любая матрица раскладывается во множество ортогональных матриц, линейная комбинация которых является достаточно точным приближением к исходной матрице.

Согласно теореме о сингулярном разложении, любая вещественная прямоугольная матрица может быть разложена на произведение трех матриц: $A = USV^T$, где матрицы U и V – ортогональные, а S – диагональная матрица сингулярных значений матрицы A.

Основная особенность сингулярного разложения заключается в том, что, согласно теореме Эккарта-Янга, если в матрице S оставить только k наибольших сингулярных значений, а в матрицах U и V — только соответствующие этим значениям столбцы, то произведение получившихся матриц S, U и V будет наилучшим приближением исходной матрицы A к матрице $\widehat{A}$ ранга k :

$$\widehat{A} \approx A = USV^T. \quad (6)$$

Таким образом, мы можем сократить исходное пространство фактически до любого размера, регулируя гиперпараметр k. Основная сложность данной процедуры заключается в нахождении баланса между сохраняемой дисперсией распределения, часть которой жертвуется для того, чтобы сократить время обучения модели и размер потребляемой памяти на хранение и использование исходного пространства признаков.

В задачах машинного обучения, связанных с обработкой естественного языка, зачастую пространство признаков расширяют различными эвристическими статистиками, так как они могут скрывать в себе важную информацию об объекте, которую крайне сложно выявить алгоритму на основе векторных значений описаний. К подобным признакам относят длину текста, отношение длины запроса к длине заголовка и так далее. Количество подобных признаков, как правило, не сильно влияет на скорость обучения модели, так как их число практически всегда остается крайне малым в сравнении с числом признаков, полученных из слов, которое, в свою очередь, может достигать сотен тысяч.

Таким образом, благодаря рассмотренным процедурам обработки текста и генерации признаков, можно добиться высокого качества построения модели и ее быстрой сходимости.

Литература

1. *Shearer C.* The CRISP-DM model: the new blueprint for data mining, 2000.
2. *Evgeniy Gabrilovich, Shaul Markovitch*. Feature generation for text categorization using world knowledge. 2005.
3. *Jan Busta, Dana Hlavackova, Milos Jakubicek, and Karel Pala*. Classification of errors in text.
4. *Dawson J. L*. (1974); Suffix Removal for Word Conflation, Bulletin of the Association for Literary and Linguistic Computing.
5. *Ms. Anjali Ganesh Jivani.* A comparative study of stemming algorithms.
6. Lemmatisation - wikipedia, the free encyclopedia. https://en.wikipedia. org/wiki/Lemmatisation.
7. *Vimala Balakrishnan and Ethel Lloyd-Yemoh* Stemming and lemmatization: A comparison of retrieval performances. IACSIT, 2014.
8. Feature extraction - wikipedia, the free encyclopedia. https://en. wikipedia.org/wiki/Feature_extraction.
9. Шумовые слова - Википедия. https://ru.wikipedia.org/wiki/Шумовые_слова.
10. *Pablo A. Estevez, Michel Tesmer Claudio A. Perez, and Jacek M. Zurada* Normalized mutual information feature selection. IEEE, 2009.