Денисов Д.В. Анализ методов машинного обучения для тематической классификации текстов // Международный журнал информационных технологий и энергоэффективности. – 2024. – Т. 9 № 4(42) с. 5–11

Международный журнал информационных технологий и энергоэффективности

Сайт журнала:
http://www.openaccessscience.ru/index.php/ijcse/

УДК 004.8

АНАЛИЗ МЕТОДОВ МАШИННОГО ОБУЧЕНИЯ ДЛЯ ТЕМАТИЧЕСКОЙ КЛАССИФИКАЦИИ ТЕКСТОВ

Денисов Д.В.
ФГАОУ ВО "НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ "МИСИС", Москва, Россия (119049, город Москва, Ленинский пр-кт, д. 4стр1), e-mail: m2205669@edu.misis.ru

В контексте растущего объема информации в цифровом пространстве, анализ и обработка текстовых данных приобретают стратегическое значение для корпоративных структур и академических групп. Сегодня текстовые данные оказывают неоспоримое влияние на процессы принятия решений, автоматизацию и стимулирование внедрения инноваций в различных отраслях. В статье приводится использование и оценка современных методов обработки и классификации текстов в базе знаний для выявления их тематической направленности.

Ключевые слова: Классификация, машинное обучение, классификация текста, анализ естественного языка, предобработка текста.

ANALYSIS OF MACHINE LEARNING METHODS FOR THEMATIC CLASSIFICATION OF TEXTS

Denisov D.V.
NATIONAL UNIVERSITY OF SCIENCE AND TECHNOLOGY "MISIS", Moscow, Russia (119049, Moscow, Leninsky Ave., 4bldg1), e-mail: m2205669@edu.misis.ru

In the context of the growing volume of information in the digital space, the analysis and processing of textual data is becoming strategically important for corporate structures and academic groups. Today, text data has an undeniable impact on decision making, automation, and driving innovation across industries. The article describes the use and evaluation of modern methods for processing and classifying texts in a knowledge base to identify their thematic focus.

Keywords: Classification, machine learning, text classification, natural language analysis, text preprocessing.

Введение

Алгоритмы машинного обучения не могут работать с текстом на естественном языке и для работы с ним нужна предварительная обработка, результатом которой является получение набора признаков [1]. Признаки в текстовых данных представляют собой характеристики текста, которые могут быть использованы для его классификации. Как правило, признаки извлекаются из текста с помощью различных методов обработки естественного языка (Natural Language Processing, NLP). Эти методы могут включать в себя токенизацию, лемматизацию, удаление стоп-слов, выделение ключевых слов и фраз, выделение частей речи и т.д.

Полученные признаки могут быть как качественными (например, наличие определенного слова или фразы), так и количественными (например, частота встречаемости слова в тексте).

Классификация — это контролируемая задача в машинном обучении, цель которой — изучить сопоставление входных объектов с выходными метками на основе набора обучающих данных. Задача классификации — это задача отнесения объекта к одному из заранее определенных классов на основании его формализованных признаков. Каждый из объектов в этой задаче представляется в виде вектора в N-мерном пространстве, каждое измерение в котором представляет собой описание одного из признаков объекта.

В области машинного обучения, задача классификации относится к категории обучения с учителем, где объекты обучающей выборки разделены на заранее определенные классы.

Методы классификации позволяют получить вероятностные характеристики для системы поддержки принятия решений. Модель классификации могут сформировать такие методы как: логистическая регрессия, дерево решений, метод опорных векторов, k-ближайших соседей, наивный байесовский метод, случайный лес, нейронные сеты и т. д.

Данная статья представляет собой анализ применения и результатов использования алгоритмов машинного обучения для классификации текстов.

Предварительная обработка текстовых данных

На начальном этапе анализа текстовые данные часто состоят из разнообразных элементов, которые могут препятствовать структурированному представлению информации. Этот этап предварительной обработки включает в себя удаление нежелательных компонентов, включая теги HTML, специальные символы, числовые значения, а также стандартизацию текста посредством преобразования регистра и других манипуляций, направленных на приведение текстовых данных к более чистой форме для последующего анализа.

После процесса очистки данных текст необходимо преобразовать в набор отдельных компонентов, известных как токены. Это подразумевает сегментирование текста на отдельные слова, символы или фразы, которые затем можно использовать для последующего анализа. Каждый токен служит независимой элементарной единицей текста, облегчающей дальнейший анализ.

При токенизации текст подвергается процедуре, направленной на удаление формы из слова и связанной с ним информации. Этот процесс, называемый лемматизацией, преобразует слова в их базовые формы, позволяя проводить анализ, основанный на их внутреннем значении. В качестве альтернативного решения также применяется стемминг, который предполагает сокращение слов до их корня (стема), игнорируя лингвистические нюансы и семантическое содержание.

Многие системы классификации текста используют несколько простых методов предварительной обработки, таких как преобразование прописных букв в строчные и удаление стоп-слов. Однако большинство систем не используют все доступные методы предварительной обработки, и исследователи полагают, что некоторые методы могут фактически оказать негативное влияние на результаты классификации. Например, исследование Формана по метрикам выбора функций для классификации текстов [2] предполагает, что стоп-слова из-за их неоднозначной природы и частого появления не обладают различительной способностью для какого-либо конкретного класса. И наоборот, ХаКоэн-Кернер и др. [3] показали, что включение словесных униграмм, содержащих стоп-

слова, в сферу ивритско-арамейских юридических документов приводит к улучшению результатов классификации по сравнению с результатами, полученными при исключении стоп-слов из словесных униграмм».

Использование текста в качестве данных для моделей машинного обучения как правило требует преобразования его в числовые признаки. Для этого используются различные методы, такие как "мешок слов", TF-IDF (term frequency-inverse document frequency), word embeddings.

Процесс векторизации текста "мешок слов" заключается в том, что каждое уникальное слово в тексте кодируется в виде отдельной размерности вектора, а сам текст формируется в виде вектора, где каждая размерность соответствует количеству употреблений соответствующего слова в тексте [4]. Таким образом, каждый текст представляется в виде вектора фиксированной длины, где каждая размерность соответствует отдельному слову из словаря. Одним из основных применений этого метода является построение моделей машинного обучения. Также этот метод может быть использован для анализа тональности текста, определения тематики текста и автоматического реферирования.

К преимуществам метода векторизации текста "мешок слов" относятся его простота и универсальность. Данный подход не требует сложных вычислений и может быть применен к любому тексту независимо от его содержания и структуры. Кроме того, этот метод позволяет эффективно работать с большими объемами текстовой информации.

Процесс векторизации текста с использованием метода TF-IDF начинается с подсчета частоты встречаемости каждого слова в каждом документе коллекции. Затем вычисляется обратная частота встречаемости слова во всех документах коллекции. Итоговый вектор для каждого документа представляет собой комбинацию значений TF и IDF для каждого слова, присутствующего в документе. TF (Term Frequency) отражает частоту встречаемости слова в документе и вычисляется как отношение числа вхождений слова к общему числу слов в документе. IDF (Inverse Document Frequency) отражает уникальность слова и вычисляется как логарифм отношения общего числа документов к числу документов, содержащих данное слово.

Метод TF-IDF позволяет выделить ключевые слова и фразы в тексте, учитывая их значимость и уникальность для каждого документа. Это делает его полезным инструментом для решения задач анализа текстовых данных, таких как поиск информации, категоризация документов, автоматическое извлечение ключевых слов и тематическое моделирование. Кроме того, метод TF-IDF может быть использован для оценки сходства между текстовыми документами, что позволяет проводить поиск и кластеризацию документов на основе их содержания.

Таким образом были рассмотрены методы предварительной обработки текстовых данных и методы создания признаков для моделей машинного обучения, особенности их применения, сильные и слабые стороны.

Методы классификации текстовых данных

Логистическая регрессия — это статистический метод, используемый для моделирования взаимосвязи между бинарной зависимой переменной и одной или несколькими независимыми переменными. Независимые переменные, также известные как предикторы или признаки, могут быть непрерывными, категориальными или сочетанием того и другого. Логистическая функция преобразует любое действительное число в диапазон [0, 1],

что делает его пригодным для моделирования вероятностей. К положительным сторонам алгоритма можно отнести в простоте интерпретации результатов. К недостаткам – линейное влияние признаков на независимую переменную и требование, чтобы каждая точка данных была независимой [5].

Дерево решений представляет собой прозрачный механизм, который позволяет пользователям легко следить за древовидной структурой и видеть, как принимается решение. Это структура, подобная блок-схеме, в которой каждый внутренний узел представляет проверку атрибута, каждая ветвь представляет результат проверки, а каждый листовой узел представляет метку класса или числовое значение. Обычно все алгоритмы дерева решений строятся в два этапа:

- рост дерева, в котором обучающий набор, основанный на локальных оптимальных критериях, рекурсивно разбивается до тех пор, пока большая часть записей, принадлежащих разделу, не будет иметь одну и ту же метку класса;
- обрезка дерева, в котором размер дерева уменьшен, что облегчает понимание.

Деревья решений популярны благодаря своей интерпретируемости, поскольку их можно легко визуализировать.

Метод опорных векторов — это мощный и универсальный алгоритм, который особенно эффективен в многомерных пространствах. SVM работает путем поиска гиперплоскости, которая лучше всего разделяет классы в пространстве объектов, с целью максимизировать разницу между классами. Опорные вектора— это точки данных, которые лежат ближе всего к поверхности решения [6]. Основным преимуществом SVM является его способность решать широкий спектр задач классификации, включая задачи большой размерности, которые не являются линейно разделимыми. Одним из основных недостатков SVM является то, что для достижения отличных результатов классификации требуется правильная установка ряда ключевых параметров.

В методе *K-ближайших соседей* (KNN) ближайший сосед измеряется относительно значения k, которое определяет, сколько ближайших соседей необходимо проверить, чтобы описать класс выборки точки данных. Одним из основных преимуществ метода KNN является то, что он эффективен для больших обучающих данных и устойчив к зашумленным обучающим данным.

Наивный байесовский метод - алгоритм вероятностной классификации, основанный на теореме Байеса с «наивным» предположением о независимости между признаками. Он часто используется в классификации текста, фильтрации спама и системах рекомендаций. Наивный алгоритм Байеса основан на самоопределяющихся предположениях. Эти гипотезы явно во многом реализуются в NLP с различными вариациями, обеспечивающим текстовую, семантическую, синтаксическую и рациональную схему [7]. К положительным сторонам относится простота интепретации, возможность использовать большие выборки и проводить мультиклассовую классификацию. К недостаткам относится то, что не всегда выполняется предположение о независимости характеристик.

Сравнение и анализ алгоритмов

В данной работе использовались следующие, реализованные на языке программирования Python, алгоритмы: дополняющий наивный байесовский классификатор, метод опорных векторов, дерево решений, случайный лес, логистическая регрессия.

В качестве набора данных выступали вопросы и ответы, взятые из базы знаний и пред обработанные с использованием мешка слов и TF-IDF.

В качестве зависимой переменной рассматривался идентификационный номер тематики в базе знаний. В качестве метрики оценки качества классификации рассматривались метрики матрицы ошибок: Accuracy, Precision, Recall и F1-Score.

Оценка качества классификации номера сообщества при использовании мешка слов представлена в Таблице 1.

Таблица 1 – Результаты классификации при использовании метода "Мешок слов"

Метрики/ Методы	**Accuracy**	**Precision**	**Recall**	**F1**
Logistic Regression	0.90	0.90	0.90	0.89
kNN	0.75	0.80	0.75	0.76
Random Forest	0.70	0.68	0.70	0.65
Complement naive Bayes	0.84	0.85	0.84	0.87
SVM	0.87	0.88	0.87	0.87

Как видно по результатам, было определено несколько методов, чье значение метрик качества классификации превышает 0.85. К ним относятся метод опорных векторов и дополняющий наивный байесовский классификатор. При этом наилучшие результаты был получены с использованием логистической регрессии

Оценка качества классификации номера сообщества при использовании TF-IDF представлена в Таблице 2.

Таблица 2 – Результаты классификации при использовании метода "TF-IDF"

Метрики/ Методы	**Accuracy**	**Precision**	**Recall**	**F1**
Logistic Regression	0.75	0.72	0.75	0.70
kNN	0.62	0.88	0.62	0.68
Random Forest	0.63	0.60	0.63	0.55
Complement naive Bayes	0.87	0.87	0.87	0.86

SVM	0.84	0.84	0.84	0.82

В сравнении с “Мешком слов” использование TF-IDF позволило получить как менее значительные показатели как точности, так и полноты. При этом методами, показавшими лучший результат, являются метод опорных векторов и дополняющий наивный байесовский классификатор, которые получили результаты, идентичные использованию метода “Мешок слов”.

Таким образом были определены методы классификации, позволяющие получить более точный и полный результат при использовании различных методов предобработки текстовых данных.

Список литературы

1. Акжолов Р.К., Верига А.В. ПРЕДОБРАБОТКА ТЕКСТА ДЛЯ РЕШЕНИЯ ЗАДАЧ NLP// Вестник науки. 2020. №3 (24). URL: https://cyberleninka.ru/article/n/predobrabotka-teksta-dlya-resheniya-zadach-nlp (дата обращения: 11.11.2023)
2. HaCohen-Kerner Y., Rosenfeld A., Sabag A., & Tzidkani M. Topic-based classification through unigram unmasking. Procedia Computer Science, - 2018. 126, pp. 69–76.
3. Song F., Liu S., & Yang J. A comparative study on text representation schemes in text categorization. Pattern analysis and applications. – 2021. 8(1–2), pp. 199–209.
4. Sebastiani F. Machine learning in automated text categorization. ACM computing surveys (CSUR). – 2002. 34(1), pp. 1–47.
5. Huang, K. Unconstrained Smartphone Sensing and Empirical Study for Sleep Monitoring and Self-Management. Ph.D. Thesis, University of Massachusetts Lowell. - 2015.
6. Y. Ke, M. Hagiwara, An English neural network that learns texts, finds hidden knowledge, and answers questions, J. Artif. Intell. Soft Comput. Res., 7 (4) (2017), pp. 229-242
7. S. Vijayarani, M.J. Ilamathi, M. Nithya, Preprocessing techniques for text mining-an overview Int. J. Comput. Sci. Commun. Networks, 5 (1) (2015), pp. 7-16

References

1. Akzholov R.K., Veriga A.V. TEXT PREPROCESSING FOR SOLVING NLP PROBLEMS. 2020. №3 (24). URL: https://cyberleninka.ru/article/n/predobrabotka-teksta-dlya-resheniya-zadach-nlp (accessed: 11.11.2023)
2. HaCohen-Kerner Y., Rosenfeld A., Sabag A., & Tzidkani M. Topic-based classification through unigram unmasking. Procedia Computer Science, - 2018. 126, pp. 69–76.
3. Song F., Liu S., & Yang J. A comparative study on text representation schemes in text categorization. Pattern analysis and applications. – 2021. 8(1–2), pp. 199–209.
4. Sebastiani F. Machine learning in automated text categorization. ACM computing surveys (CSUR). – 2002. 34(1), pp. 1–47.
5. Huang, K. Unconstrained Smartphone Sensing and Empirical Study for Sleep Monitoring and Self-Management. Ph.D. Thesis, University of Massachusetts Lowell. - 2015.
6. Y. Ke, M. Hagiwara, An English neural network that learns texts, finds hidden knowledge, and answers questions, J. Artif. Intell. Soft Comput. Res., 7 (4) (2017), pp. 229-242

7. S. Vijayarani, M.J. Ilamathi, M. Nithya, Preprocessing techniques for text mining-an overview Int. J. Comput. Sci. Commun. Networks, 5 (1) (2015), pp. 7-16