Предобработка и классификация текста

Классификация текста – задача компьютерной лингвистики, заключающаяся в отнесении документа к одной из нескольких категорий на основании содержания документа. Классификация текстов как правило применяется для разделения сайтов по тематическим каталогам, борьбы со спамом, распознавания эмоциональной окраски текстов, персонификации рекламы и т. д. В данной работе рассматривается именно тематическая каталогизация.

Формализовано задачу классификации можно поставить следующим образом:

- X – множество описаний объектов,
- Y – конечное множество номеров (имён, меток) классов,
- $X^m = \{(x_1, y_1), \dots, (x_m, y_m)\}$ – конечная обучающая выборка,
- y^* - целевая зависимость, отображение значения которой известны только на объектах обучающей выборки.

Требуется построить алгоритм $\alpha: X \to Y$, способный классифицировать произвольный объект $x \in X$.

В задаче агрегации новостных статей необходимо провести классификацию документов для определения тематики содержания. В общей сложности было выявлено и размечено 7 классов.

Предобработка текста

Для очистки документа от возможных шумов и аномалий, которые могут помешать построению достоверного вектора, производится предобработка текста. Предобработка включает в себя: приведение слов к нижнему регистру, токенизацию, удаление стоп-слов, стемминг и лемматизацию.

Приведение слов к нижнему регистру необходимо для устранения помех, связанных с расстановкой одинаковых слов в начале и середине предложения, что меняет их регистр написания.

Токенизация- процесс разбиения текста на текстовые единицы, например, слова или предложения.

Удаление стоп-слов- извлечение слов с большой частотой встречаемости, которые не несут большой смысловой нагрузки.

Стемминг- нахождения основы слова.

Лемматизация- процесс приведения слова к нормальной (словарной) форме.

После предобработки документов необходимо выбрать алгоритм векторизации текста. В данной работе будут рассматриваться несколько векторизаторов, а именно: TF-IDF, Word2Vec, Doc2Vec, FastText, GloVe, Universal-Sentence-Encoder и Bert. Будут рассмотрены принципы их работы и произведено сравнение их влияния на точность классификаторов. Суть векторизации документов заключается в построении векторов для каждого текста. Близкие по смыслу с точки зрения модели документы (похожие по смыслу слова используются в сходных контекстах) будут близки по косинусной мере.

Косинусное сходство- это мера сходства между двумя векторами, которая используется для измерения косинуса угла между ними.

Сверточные нейронные сети (CNN)

Это класс глубоких нейронных сетей, которые особенно эффективны для обработки данных с сетчатой структурой.

CNN состоит из нескольких типов слоев, каждый из которых выполняет свою функцию:

1. Сверточные слои (Convolutional Layers):
   Основной компонент CNN. Они применяют фильтры (или ядра свертки) к входным данным для извлечения признаков. Каждый фильтр сканирует изображение, вычисляя свертку, которая создает карту признаков (feature map).
   Свертка помогает выявить локальные зависимости и иерархию признаков.
2. Слои подвыборки (Pooling Layers):
   Обычно используются после сверточных слоев для уменьшения пространственного разрешения карты признаков и снижения вычислительной нагрузки. Наиболее распространены операции максимального (max pooling) и среднего (average pooling) подвыборки.
   Это также помогает сделать модель более устойчивой к незначительным изменениям в изображениях.
3. Полносвязные слои (Fully Connected Layers):
   Находятся в конце сети и соединяют все нейроны предыдущего слоя с каждым нейроном текущего слоя. Эти слои помогают объединить извлеченные признаки для принятия окончательного решения (например, классификации).
4. Активационные функции:
   Используются после сверточных и полносвязных слоев для введения нелинейности в модель. Наиболее популярной является функция ReLU (Rectified Linear Unit), но также могут использоваться другие функции, такие как sigmoid или tanh.

Преимущества CNN:

- **Автоматическое извлечение признаков:** CNN автоматически учит признаки на разных уровнях абстракции, что снижает необходимость ручного отбора признаков.
- **Параметрическая эффективность:** Использование сверток позволяет значительно сократить количество параметров по сравнению с полносвязными сетями.
- **Устойчивость к смещениям:** За счет применения локальных фильтров и подвыборки модели становятся более устойчивыми к небольшим изменениям и шуму в данных.

Рекуррентные нейронные сети (RNN)

Это класс нейронных сетей, специально разработанных для обработки последовательных данных. Они широко используются в задачах, связанных с временными рядами, текстом, аудио и другими последовательными структурами, где важен порядок элементов.

Основная идея RNN заключается в том, что они имеют внутреннее состояние (или память), которое позволяет им сохранять информацию о предыдущих входах. Это достигается за счет рекуррентных связей, которые позволяют передавать информацию от одного шага времени к следующему.

Входные данные: На каждом временном шаге t RNN принимает входной вектор x_t.

Скрытое состояние: Скрытое состояние h_t обновляется на основе текущего входа и предыдущего скрытого состояния:

$$h_t = f(W_h h_{t-1} + W_x x_t + b)$$

где W_h и W_x — это веса для скрытого состояния и входа соответственно, b — смещение, а f — активационная функция (обычно используется tanh или ReLU).

Выход: На каждом временном шаге может быть сгенерирован выходной вектор y_t:

$$y_t = W_y h_t + b_y$$

где W_y — это веса для выхода, а b_y — смещение.

Проблемы RNN

Несмотря на свою эффективность в работе с последовательными данными, классические RNN имеют некоторые ограничения:

Затухание градиента: При обучении RNN через обратное распространение по времени (BPTT) градиенты могут затухать или взрываться, что затрудняет обучение долгосрочных зависимостей.

Краткосрочная память: Классические RNN лучше работают с краткосрочными зависимостями и могут не справляться с долгосрочными.