Обработка естественного языка. Лекция 1
Введение и классические алгоритмы

Михаил Пожидаев

6 сентября 2024 г.

Классические задачи

1. Определение части речи (*part of speech*, *POS*).
2. Построение дерева зависимости слов в предложении (обычно начиная с подлежащего).
3. Морфологическое аннотирование (тегирование).

Токенизация

Токенизация — преобразование входной строки
для выделения групп символов, формирующих токены.

1. Точные требования к токену определяются задачей (включение пробелов, дробные части слов и пр.).
2. Иногда проводится аннотирование и распознавание понятий (цен, дат, инициалов и пр.).
3. В большинстве случаев является первым этапом обработки текста.

Сегментация

Сегментация — разделение текста (или последовательности токенов) на синтаксически обособленные группы (между сегментами нет синтаксических связей, только семантические).

1. Появление точки, вопросительного знака или восклицательного знака с последующим словом, начинающимся с заглавной буквы, — необходимое условие начала нового сегмента (предложения), но не достаточное!
2. Сегментация и токенизация в общем случае являются многофакторной задачей, требующей учёта множества различных аспектов (которые даже человеку не всегда легко разобрать).

Подготовка слов

После токенизации и сегментации

Промежуточный этап обработки текста, который в ряде задач проводится после токенизации — это лемматизация слов (lemmatization).

Начальная форма слова

Под леммой подразумевается начальная или словарная форма слова, в именительном падеже и в единственном числе для имён существительных, инфинитив для глаголов и т. д.

Это делается для того, чтобы токены обозначали не конкретное слово, а лексему, которой это слово принадлежит.

Применение

Улучшим поиск текста

Запрос «Пушкин» обозначает имя писателя во всех вариантах его использования, такому запросу должны соответствовать разные фразы, включая «имение Пушкина», «воспоминания о Пушкине» и пр.

Поисковый индекс

Индекс, на основе которого производится отбор документов, включает не сами исходные слова, а их леммы («имение Пушкин», «воспоминание о Пушкин»).

Неоднозначность ответа

Множественность результатов лемматизации

Результат лемматизации неоднозначен

Точнее, он чаще всего представлен не одним, а несколькими словами. слово «веках» может иметь леммы «век» и «веко».

Таким образом, если разработчик не хочет потерять полноту поиска, ему придётся сохранять в поисковом индексе все леммы, которые могут быть получены для каждого слова.

Стемминг

Поиск основы слова

С задачей лемматизации близко связана задача стемминга (stemming) — сохранение основы слова, т. е. исходного слова с отброшенными приставкой и окончанием. лемматизация — это стемминг с присоединением умолчательного окончания.
С функциональной точки зрения различия между стеммингом и лемматизацией минимальны.

Почти компилятор

Распознающие формальные грамматики

1. Задаются алфавит, множество терминальных символов и нетерминальных символов.
2. Задаётся множество правил, у которых в левой части находится один нетерминал, а в правой последовательность из терминальных и нетерминальных символов.
3. Задаётся корневой нетерминальный символ.

Требуется определить дерево правил, терминальные символы которых совпадают с распознаваемой последовательностью.

Томита-парсер

Томита-парсер (*Tomita-parser*) — открытая утилита компании Яндекс для обработки естественного языка.

1. Производит извлечение фактов из текста.
2. Текст анализируется посредством распознающих формальных грамматик.
3. Использование затруднено большой трудоёмкостью обработки текста при недетерминированных правилах и сложностью расстановки приоритетов.

Термины

Основные понятия Томита-парсера

1. Газеттир — множество контекстносвободных правил.
2. Факт — запись с атрибутами, описывающая распознанное понятие.
3. Грамматика — способ связи правил и фактов.

Пример грамматики

```
Ent -> 'свод' 'череп' ;
Ent -> Adj<gnc-agr[1]>+ Noun<gnc-agr[1], rt> ;
Rel -> Ent interp (Equality.Name1) "являться"\
     Ent interp (Equality.Name2) ;
Rel -> Ent interp (Equality.Name1) "образовывать"\
     Ent interp (Equality.Name2) ;
```

TF-iDf

Term freq — inverse document freq

TF-iDF — статистическая мера, характеризующая вес лексемы в некотором документе в составе множества документов. Является хорошей характеристикой для лексического анализа текстов и корпусов текстов.

$$tfidf_{ij} = tf_{ij}\, idf_{ij}$$

$$tf_{ij} = \frac{c_{ij}}{n_j}$$

$$idf_{ij} = \log \frac{p}{d_i}$$

Обозначения:

- i — анализируемая лексема, j документ в корпусе документов;
- c_{ij} — количество употреблений лексемы в документе, n_j — общее количество всех лексем в документе;
- p — общее количество документов, d_i — количество документов, содержащий лексему i.

Применение СММ

Скрытые марковские модели применяются для решения следующих задач в лингвистике:

1. Определение частей речи.
2. Распознавание речи (один из первых применённых алгоритмов).
3. Восстановление повреждённого текста.

На практике

СММ в дискретном времени

Заданы множества скрытых состояний $X = \{x_1, x_2, \ldots, x_n\}$, множество наблюдаемых состояний $Y = \{y_1, y_2, \ldots, y_m\}$ и множество шагов времени $\{t_1, t_2, \ldots, t_l\}$ (*timestep*).
Фактические состояния в моменты времени неизвестны, их следует определить, но известны наблюдаемые состояния. Также заданы матрица вероятности S_{ij} перехода из одного состояния x_i в другое, $P(y_i|x_j)$ — вероятность наблюдения y_i при скрытом состоянии x_j и z — вектор вероятности начальных состояний.

Алгоритм Витерби

Алгоритм позволяет определить последовательность скрытых состояний (путь Витерби), имеющую максимальную вероятность среди всех возможных вариантов последовательности. Порядок вычисления матрицы в алгоритме Витерби следующий:

$$V_{1k} = P(y_1|k)z_k$$
$$V_{tk} = \max_{x \in X}(P(y_t|k)S_{xk}V_{(t-1)x})$$

Спасибо за внимание!

Всё о курсе: https://marigostra.ru/materials/nlp.html
E-mail: msp@luwrain.org
Канал в Телеграм: https://t.me/MarigostraRu