**Липецкий государственный технический университет**

Факультет автоматизации и информатики

Кафедра автоматизированных систем управления

ИНДИВИДУАЛЬНОЕ ДОМАШНЕЕ ЗАДАНИЕ

по дисциплине «Прикладные интеллектуальные системы и экспертные системы»

Обработка естественного языка

Студент Крутских А.Ю.

Группа М-ИАП-22

Руководитель Кургасов В.В.

Доцент, к.п.н.

Липецк 2022 г.

Краткая теоретическая справка

Обработка естественного языка (Natural Language Processing, NLP) — пересечение машинного обучения и математической лингвистики, направленное на изучение методов анализа и синтеза естественного языка. Сегодня NLP применяется во многих сферах, в том числе в голосовых помощниках, автоматических переводах текста и фильтрации текста. Основными тремя направлениями являются: распознавание речи  (Speech Recognition), понимание естественного языка (Natural Language Understanding) и генерация естественного языка (Natural Language Generation).

NLP решает большой набор задач, который можно разбить по уровням (в скобках). Среди этих задач, можно выделить следующие:

- Распознавание текста, речи, синтез речи (сигнал);

- Морфологический анализ, канонизация (слово);

- POS-тэгирование, распознавание именованных сущностей, выделение слов (словосочетание);

- Синтаксический разбор, токенизация предложений (предложение);

Извлечение отношений, определение языка, анализ эмоциональной окраски (абзац);

- Аннотация документа, перевод, анализ тематики (документ);

- Дедубликация, информационный поиск (корпус).

Основными подходами к данной задаче являются:

1) Предобработка текста

Предобработка текста переводит текст на естественном языке в формат удобный для дальнейшей работы. Предобработка состоит из различных этапов, которые могут отличаться в зависимости от задачи и реализации. Далее приведен один из возможных набор этапов:

- Перевод всех букв в тексте в нижний или верхний регистры;

- Удаление цифр (чисел) или замена на текстовый эквивалент (обычно используются регулярные выражения);

- Удаление пунктуации. Обычно реализуется как удаление из текста символов из заранее заданного набора;

- Удаление пробельных символов (whitespaces);

- Токенизация (обычно реализуется на основе регулярных выражений);

- Удаление стоп слов;

- Стемминг;

- Лемматизация;

- Векторизация.

2) Стемминг

Количество корректных словоформ, значения которых схожи, но написания отличаются суффиксами, приставками, окончаниями и прочим, очень велико, что усложняет создание словарей и дальнейшую обработку. Стемминг позволяет привести слово к его основной форме. Суть подхода в нахождении основы слова, для этого с конца и начала слова последовательно отрезаются его части. Правила отсекания для стеммера создаются заранее, и чаще всего представляют из себя регулярные выражения, что делает данный подход трудоемким, так как при подключении очередного языка нужны новые лингвистические исследования. Вторым недостатком подхода является возможная потеря информации при отрезании частей, например, мы можем потерять информацию о части речи.

3) Лемматизация

Данный подход является альтернативой стемминга. Основная идея в приведении слова к словарной форме — лемме. Например, для русского языка:

- для существительных — именительный падеж, единственное число;

- для прилагательных — именительный падеж, единственное число, мужской род;

- для глаголов, причастий, деепричастий — глагол в инфинитиве несовершенного вида.

4) Векторизация

Большинство математических моделей работают в векторных пространствах больших размерностей, поэтому необходимо отобразить текст в векторном пространстве. Основным походом является мешок слов (bag-of-words): для документа формируется вектор размерности словаря, для каждого слова выделяется своя размерность, для документа записывается признак насколько часто слово встречается в нем, получаем вектор. Наиболее распространенным методом для вычисления признака является TF-IDF (TF — частота слова, term frequency, IDF — обратная частота документа, inverse document frequency). TF вычисляется, например, счетчиком вхождения слова. IDF обычно вычисляют как логарифм от числа документов в корпусе, разделённый на количество документов, где это слово представлено. Таким образом, если какое-то слово встретилось во всех документах корпуса, то такое слово не будет никуда добавлено. Плюсами мешка слов является простая реализация, однако данный метод теряет часть информации, например, порядок слов. Для уменьшения потери информации можно использовать мешок N-грамм (добавлять не только слова, но и словосочетания), или использовать методы векторных представлений слов — это, например, позволяет снизить ошибку на словах с одинаковыми написаниями, но разными значениями.

5) Дедубликация

Так как количество схожих документов в большом корпусе может быть велико, необходимо избавляться от дубликатов. Так как каждый документ может быть представлен как вектор, то мы можем определить их близость, взяв косинус или другую метрику. Минусом является то, что для больших корпусов полный перебор по всем документам будет невозможен. Для оптимизации можно использовать локально-чувствительный хеш, который поместит близко похожие объекты.

6) Семантический анализ

Семантический (смысловой) анализ текста — выделение семантических отношений, формировании семантического представления. В общем случае семантическое представление является графом, семантической сетью, отражающим бинарные отношения между двумя узлами — смысловыми единицами текста. Глубина семантического анализа может быть разной, а в реальных системах чаще всего строится только лишь синтаксико-семантическое представление текста или отдельных предложений. Семантический анализ применяется в задачах анализа тональности текста (Sentiment analysis), например, для автоматизированного определения положительности отзывов.

7) Распознавание именованных сущностей и извлечение отношений

Именованные сущности — объекты из текста, которые могут быть отнесены к одной из заранее заявленных категорий (например, организации, личности, адреса). Идентификация ссылок на подобные сущности в тексте является задачей распознавания именованных сущностей.

8) Распознавание именованных сущностей и извлечение отношений

Именованные сущности — объекты из текста, которые могут быть отнесены к одной из заранее заявленных категорий (например, организации, личности, адреса). Идентификация ссылок на подобные сущности в тексте является задачей распознавания именованных сущностей.

Реализация на Python

import spacy

import textacy.extract

# Загрузка английской NLP-модели

nlp = spacy.load('en\_core\_web\_lg')

# Текст для анализа

text = """London is the capital and most populous city of England and

the United Kingdom. Standing on the River Thames in the south east

of the island of Great Britain, London has been a major settlement

for two millennia. It was founded by the Romans, who named it Londinium.

"""

# Парсинг текста с помощью spaCy. Эта команда запускает целый конвейер

doc = nlp(text)

# в переменной 'doc' теперь содержится обработанная версия текста

# мы можем делать с ней все что угодно!

for entity in doc.ents:

print(f"{entity.text} ({entity.label\_})")

# Если токен является именем, заменяем его словом "REDACTED"

def replace\_name\_with\_placeholder(token):

if token.ent\_iob != 0 and token.ent\_type\_ == "PERSON":

return "[REDACTED] "

else:

return token.text

# Проверка всех сущностей

def scrub(text):

doc = nlp(text)

with doc.retokenize() as retokenizer:

for ent in doc.ents:

retokenizer.merge(ent)

tokens = map(replace\_name\_with\_placeholder, doc)

return "".join(tokens)

s = """

In 1950, Alan Turing published his famous article "Computing Machinery and Intelligence". In 1957, Noam Chomsky’s

Syntactic Structures revolutionized Linguistics with 'universal grammar', a rule based system of syntactic structures.

"""

print(scrub(s))

text = """London is the capital and most populous city of England and the United Kingdom.

Standing on the River Thames in the south east of the island of Great Britain,

London has been a major settlement for two millennia. It was founded by the Romans,

who named it Londinium.

"""

# Анализ

doc = nlp(text)

# Извлечение полуструктурированных выражений со словом London

statements = textacy.extract.semistructured\_statements(doc, entity="London", cue="be")

# Вывод результатов

print("Here are the things I know about London:")

for statement in statements:

subject, verb, fact = statement

print(f" - {fact}")

Результат выполнения программы

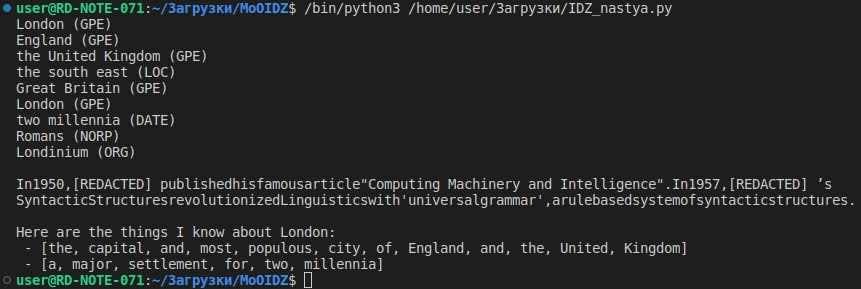


Рисунок 1 – Результат выполнения программы

Вывод

В ходе выполнения индивидуального домашнего задания были изучены теоретические основы обработки естественного языка и реализована модель NLP на языке Python.

Для этого использовалась библиотека spaCy – библиотека, разработанная по методологии SCRUM на языке Cypthon, позиционируется как самая быстрая NLP библиотека. Имеет множество возможностей, в том числе, разбор зависимостей на основе меток, распознавание именованных сущностей, пометка частей речи, векторы расстановки слов. Не поддерживает русский язык.

Подводя итоги, хочется отметить, что задача NLP может решаться в ходе, например, реализации голосового ассистента, а также когда нужно сделать выжимку из статей, не упустив основную мысль работы. В связи с этим NLP является актуальной на сегодняшний день задачей.

Список литературы

1. Статья «Обработка естественного языка» - <https://neerc.ifmo.ru/wiki/index.php?title=Обработка_естественного_языка>, свободный доступ.

2. Документация к библиотеке spaCy - <https://spacy.io/usage>, свободный доступ.