ПРИМЕНЕНИЕ МЕТОДА ВЕКТОРИЗАЦИИ ДЛЯ АНАЛИЗА РУССКОЯЗЫЧНОЙ ТЕКСТОВОЙ ИНФОРМАЦИИ

Гоглев И. В.
Кафедра информационных технологий автоматизированных систем,
Белорусский государственный университет информатики и радиоэлектороники
Минск, Республика Беларусь
E-mail: ivangoglev1998@gmail.com

Рассматривается реализация вопросно-ответной системы на основе векторизации

Введение

На сегодняшний день существует большое множество вопросно-ответных систем (QA-систем). Вопросно-ответная система – это информационная система, являющаяся гибридом поисковых, справочных и интеллектуальных систем, которая использует естественно-языковой интерфейс.[1] Различают общие QA-системы и узкоспециализированные QA-системы. В то же время развились систем и методы обработки информации на естественном языке. Для создания простой QA-системы предлагается использовать метод векторизации предложений.

I. Анализ русскоязычной текстовой информации на Python

Для анализа текстов на русском языке используется библиотека pymorphy2. Pymorphy2 — морфологический анализатор, разработанный на языке программирования Python. Выполняет лемматизацию и анализ слов, способен осуществлять склонение по заданным грамматическим характеристикам слов. Работает со словарём OpenCorpora, а для незнакомых слов строит гипотезы[2]. Ниже представлен пример работы данного анализатора.

```
In[7]: import pymorphy2
In[8]: pymorphy2.MorphAnalyzer().parse('книга')
```

Рис. 1 – Пример использования библиотеки pymorthy2 для морфологического анализа

Для каждого разбора анализатор вычисляет скоринг[2]. Скоринг разбора предсавляет собой условную вероятность тега $P(tag|word)$. Она вычисляется на основе частот словаря OpenCorpora для всех неоднозначных слов со снятой неоднозначностью.[2] Для каждого подобного слова оценивается его частота в корпусе и количество сопоставлений его тега. В результате получаем отсортированый по убыванию скоринга массив объектов класса Parse. Для слова «книга» анализатор выдаёт ответ следующего вида (рис.2).

```
[Parse(word='книга',
tag=OpencorporaTag('NOUN,inan,femn sing,nomn'),
normal_form='книга',
score=1.0,
methods_stack=((DictionaryAnalyzer(),
'книга', 44, 0),))]
```

Рис. 2 – Результат морфологического анализа

Поле tag содержит морфологическую информацию о введённом слове (таблица 1)

Таблица 1 – Теги морфологического анализа

Тег	Смысл тега
NOUN	имя существительное
inan	неодушевлённое(inanimate)
femn	женский род(feminine)
sing	единственное число(singular)
nomn	именительный падеж(nominative)

Также предусмотрен вывод морфологической информации на русском языке (рис.3).

```
In[12]: pymorphy2.MorphAnalyzer().parse('книга')[0].tag.cyr_repr
Out[12]: 'СУЩ,неод,жр ед,им'
```

Рис. 3 – Морфологический анализ на русскоом языке

MorphAnalyzer предоставляет возможность получить нормальную форму слова (именительный падеж и единственное число для существительных) (рис. 4).

```
In[14]: pymorphy2.MorphAnalyzer().parse('книга')[0].normal_form
Out[14]: 'книга'
```

Рис. 4 – Нормальная форма слова

II. Применение метода векторизации для ответа на вопрос по тексту

Перед векторизацией необходимо провести предварительную обработку текстовых данных. Текст преобразуется в массив строк (предложений) и каждое слово в строке приведено к нормальной форме. Стоп-слова (слова, которые не придают особого значения предложению) удаляются из каждого предложения(рис. 5).

```
In[15]: sentences = [
   ...:     'Он играть поле',
   ...:     'Он играть парни футбол',
   ...:     'Игра футбол закончить',
   ...:     'Дима Вася собирать цветы'
   ...: ]
In[16]: questions = ['Кто играть футбол']
```

Рис. 5 – Предварительная обработка текстовых данны

После предварительной обработки формируем датасет (массив предложений текста и вопроса) (рис. 6)

```
In[17]: sentences+questions
Out[17]:
['Он играть поле',
 'Он играть парни футбол',
 'Игра футбол закончить',
 'Дима Вася собирать цветы',
 'Кто играть футбол']
```

Рис. 6 – Сформированный датасет

Далее производим векторизацию с помощью функции CountVectorizer из библиотеки sklearn (рис. 7).

```
In[21]: from sklearn.feature_extraction.text import CountVectorizer
In[22]: dataset=sentences+questions
In[23]: vectorizer = CountVectorizer(input=dataset)
In[24]: snts_vectors = vectorizer.fit_transform(dataset)
In[25]: vectors = snts_vectors.toarray()
In[26]: vectors
Out[26]:
array([[0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0],
       [0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 1, 0],
       [0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 1, 0],
       [1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1],
       [0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 1, 0]])
```

Рис. 7 – Векторизация

Полученный результат содержит информацию о взаимосвязях между предложениями (в том числе и их подобии)[3]

Далее разбиваем результат на массив векторов предложений текста и вектор вопроса. После сравниваем векторы предложений с вектором вопроса. Предложение с наиболее близким вектором являеся ответом на вопрос. (рис. 8)

```
In[34]: print('q: ', questions[0])
   ...: if max_zero == 0:
   ...:   print('a: ', 'Этого нет в тексте')
   ...: else:
   ...:   print('a: ', sentences[resp])
   ...:
q:  Кто играть футбол
a:  Он играть парни футбол
```

Рис. 8 – Реультат работы

III. Выводы

Библиотека pymorthy2 и метод векторизации позволяет создать простую вопросно-ответную систему с возможностью предварительной обработки вводимого текста. Одно из главных преимуществ использования векторизации — относительная простота реализации. На сегодняшний день существует большое множество сторонних библиотек машинного обучения для обработки и анализа информации на естественном языке.

Используя метод векторизации возможно создать большое множество различных узкоспециализированных вопросно-ответных систем.

IV. Список литературы

1. Gigabaza.ru [Электронный ресурс]. – Режим доступа: https://gigabaza.ru/doc/67598.html. – Дата доступа: 21.10.2021.
2. Nlpub.ru [Электронный ресурс]. – Режим доступа: https://nlpub.ru/Pymorphy. – Дата доступа: 21.10.2021.
3. Хобсон, Л. Обработка естественного языка в действии / Л. Хобсон, Х. Ханнес, Х. Коул. – СПб // Питер. – 2020. – С.575