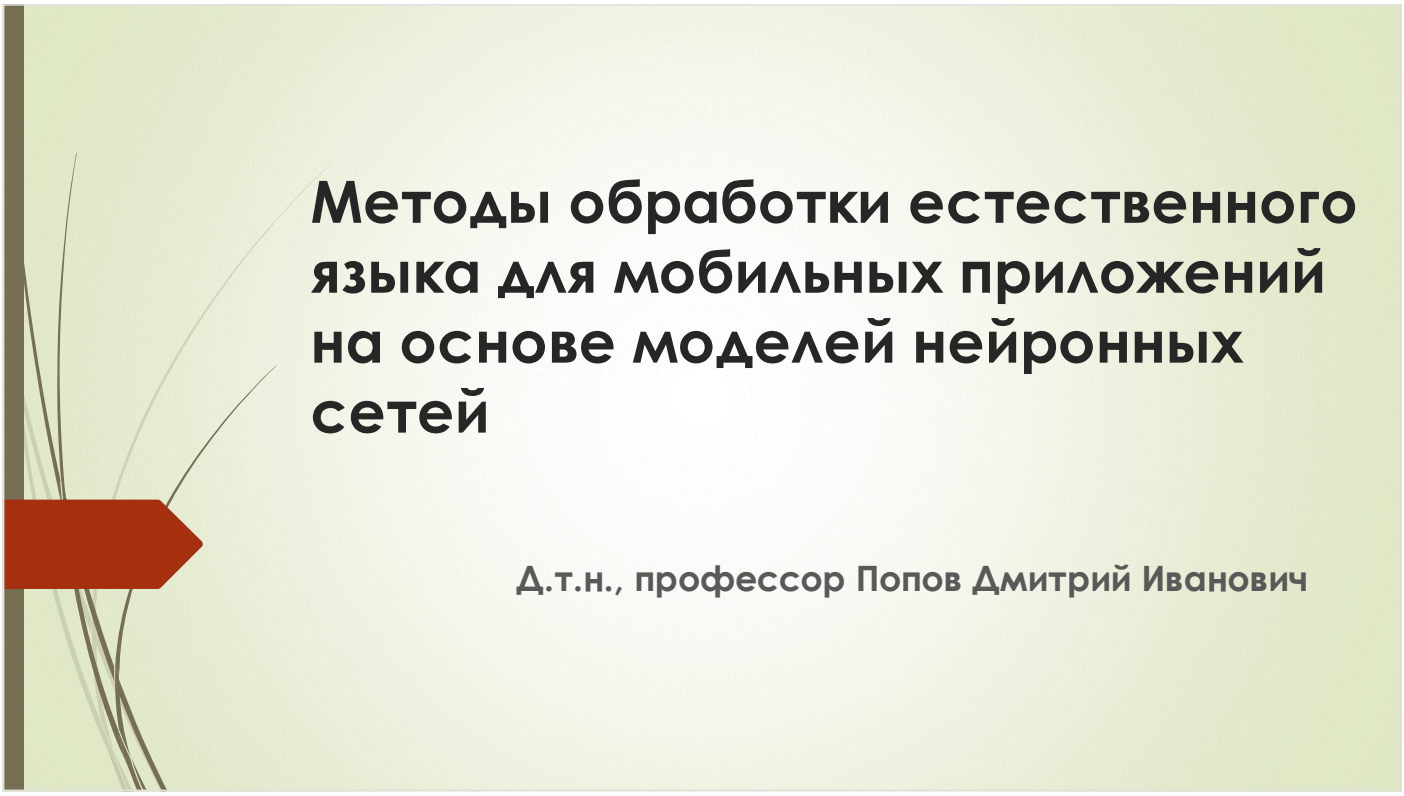

Методы обработки естественного языка для мобильных приложений на основе моделей нейронных сетей
Д.т.н., профессор Попов Дмитрий Иванович

2

Основные задачи к проработке на базе методов обработки естественного языка (Nature Language Processing - NLP)

- Ввод ответа человека с микрофона в виде звукового файла
- Преобразование(распознавание) речи в текстовый файл. Используются библиотеки SpeechRecognition, gTTS, PyAudio.
- Обработка ввода в виде текста
- Генерация ответа

3

Обработка ввода в виде текста
Этап 1. Очистка ввода

- Удаление лишних пробелов, случайных символов и т.п. Удалить все нерелевантные символы (например, любые символы, не относящиеся к цифро-буквенным).
- ***Токенизировать*** текст, разделив его на индивидуальные слова.
- Удалить нерелевантные слова — например, @vasya_pupkin (упоминания в Twitter) или URL-ы и т.п. **Удаляем стоп-слова** из нашего текста, поскольку в случае анализа настроений стоп-слова могут не содержать никакой полезной информации.
- Перевести все символы в нижний регистр для того, чтобы слова «привет», «Привет» и «ПРИВЕТ» считались одним и тем же словом.
- Рассмотрите возможность совмещения слов, написанных с ошибками, или имеющих альтернативное написание (например, «круто»/«круть»/ «крууууто»). Работа с опечатками осуществляется с использованием словарей и/или учета расстояния Левенштейна при поиске и т.п.
- Под расстоянием Левенштейна понимается минимальное количество операций удаления, вставки и замены символа, необходимое для преобразования одной строки (введенной с опечатками) в другую (эталонная, из базы данных/репозитория строк). Этот этап в англоязычной литературе часто называется ***spell correction***.

4

Токенизация

Токенизация разделяет большое количество текста на более мелкие фрагменты, известные как токены. Эти фрагменты или токены очень полезны для поиска закономерностей и рассматриваются в качестве основного шага для ***стемминга*** и ***лемматизации***. Токенизация также поддерживает замену конфиденциальных элементов данных на нечувствительные.

5

Пример Токенизации. Функция split()

```
my_text="""Токенизация разделяет большое количество текста на более мелкие
фрагменты, известные как токены. Эти фрагменты или токены очень полезны для
поиска закономерностей и рассматриваются в качестве основного шага для
стемминга и лемматизации. Токенизация также поддерживает замену
конфиденциальных элементов данных на нечувствительные.
"""
print(my_text.split())
```

```
['Токенизация', 'разделяет', 'большое', 'количество', 'текста', 'на', 'более'
'мелкие', 'фрагменты,', 'известные', 'как', 'токены.', 'Эти', 'фрагменты',
'или', 'токены', 'очень', 'полезны', 'для', 'поиска', 'закономерностей', 'и'
'рассматриваются', 'в', 'качестве', 'основного', 'шага', 'для', 'стемминга',
'и', 'лемматизации.', 'Токенизация', 'также', 'поддерживает', 'замену',
'конфиденциальных', 'элементов', 'данных', 'на', 'нечувствительные.']
```

6

Пример Токенизации. Библиотека nltk

```
import nltk
nltk.download('punkt')

from nltk.tokenize import word_tokenize

print(word_tokenize(my_text))
```

```
['Токенизация', 'разделяет', 'большое', 'количество', 'текста', 'на', 'более',
'мелкие', 'фрагменты', ',', 'известные', 'как', 'токены', '.', 'Эти',
'фрагменты', 'или', 'токены', 'очень', 'полезны', 'для', 'поиска',
'закономерностей', 'и', 'рассматриваются', 'в', 'качестве', 'основного',
'шага', 'для', 'стемминга', 'и', 'лемматизации', '.', 'Токенизация',
'также', 'поддерживает', 'замену', 'конфиденциальных',
'элементов', 'данных', 'на', 'нечувствительные', '.']
```

Поиск стоп-слов в nltk

7

```
nltk.download('stopwords')
```

```
[nltk_data] Downloading package stopwords to
/root/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
True
```

```
from nltk.corpus import stopwords

stopwords_ru = stopwords.words("russian")

print(stopwords_ru)
```

```
['и', 'в', 'во', 'не', 'что', 'он', 'на', 'я', 'с', 'со', 'как', 'а', 'то', 'все', 'она', 'так',
'его', 'но', 'да', 'ты', 'к', 'у', 'же', 'вы', 'за', 'бы', 'по', 'только', 'ее', 'мне', 'было',
'вот', 'от', 'меня', 'еще', 'нет', 'о', 'из', 'ему', 'теперь', 'когда', 'даже', 'ну', 'вдруг',
'ли', 'если', 'уже', 'или', 'ни', 'быть', 'был', 'него', 'до', 'вас', 'нибудь', 'опять', 'уж',
'вам', 'ведь', 'там', 'потом', 'себя', 'ничего', 'ей', 'может', 'они', 'тут', 'где', 'есть',
'надо', 'ней', 'для', 'мы', 'тебя', 'их', 'чем', 'была', 'сам', 'чтоб', 'без', 'будто', 'чего',
'раз', 'тоже', 'себе', 'под', 'будет', 'ж', 'тогда', 'кто', 'этот', 'того', 'потому', 'этого',
'какой', 'совсем', 'ним', 'здесь', 'этом', 'один', 'почти', 'мой', 'тем', 'чтобы', 'нее', 'сейчас',
'были', 'куда', 'зачем', 'всех', 'никогда', 'можно', 'при', 'наконец', 'два', 'об', 'другой',
'хоть', 'после', 'над', 'больше', 'тот', 'через', 'эти', 'нас', 'про', 'всего', 'них', 'какая',
'много', 'разве', 'три', 'эту', 'моя', 'впрочем', 'хорошо', 'свою', 'этой', 'перед', 'иногда',
'лучше', 'чуть', 'том', 'нельзя', 'такой', 'им', 'более', 'всегда', 'конечно', 'всю', 'между']
```

Лемматизая, удаление стоп-слов в nltk

8

```
import re
from pymorphy2 import MorphAnalyzer
from nltk.corpus import stopwords

patterns = "[A-Za-z0-9!#$%&'()*+,./:;<=>?@[\]^_`{|}~—\"\-]+"
stopwords_ru = stopwords.words("russian")
morph = MorphAnalyzer()

def lemmatize(doc):
    doc = re.sub(patterns, ' ', doc)
    tokens = []
    for token in doc.split():
        if token and token not in stopwords_ru:
            token = token.strip()
            token = morph.normal_forms(token)[0]
            tokens.append(token)
    if len(tokens) > 2:
        return tokens
    return None
```

9

Обработка ввода в виде текста
Этап 2. Лемматизация

- Это процесс преобразования слова в его базовую/словарную форму (избавление от спряжений, склонений, множественных форм и т.д.). Например, «машина» вместо «машиной», «на машине», «машинах» и пр.
- Упрощенная форма лемматизации – ***стемминг***. Это когда в слове удаляются последние несколько символов, связанные с окончанием и/или суффиксом, оставляя корень слова.
- Однако такое упрощение нужно использовать с осторожностью, т.к. это может привести к некоторым ошибкам, например, «Caring» -> Лемматизация -> «Care» (заботиться), «Caring» -> Стемминг -> «Can»(машина).
- Элементы лемматизации на русском языке есть в открытой Библиотеке Natasha.

10

Обработка ввода в виде текста
Этап 3. Классификация намерений

- Этот этап позволяет выявить в фразах пользователя его намерения, желания, вопросы и в соответствии с этими ожиданиями пользователя вести диалог.
- В простейшем случае составляется словарь, который сопоставляет некоторые типовые фразы, словосочетания, слова множеству возможных намерений.
- В англоязычной литературе это - **intent classification, intent extraction**

11

Обработка ввода в виде текста
Этап 4. Извлечение сущностей

- Одной из самых сложных и неоднозначных проблем, которая может встретиться во время работы с данными, является извлечение именованных сущностей (Named-entity recognition, NER) – слов, обозначающих предмет или явление определенной категории.
- Для этого может быть использована библиотека Natasha. Библиотека Natasha решает базовые задачи обработки ***естественного русского языка***: сегментация на токены и предложения, морфологический и синтаксический анализ, лемматизация, извлечение именованных сущностей. Для новостных статей качество на всех задачах сравнимо или превосходит существующие решения. Библиотека поддерживает Python 3.5+ и PyPy3, не требует GPU, зависит только от NumPy.

12

Обработка ввода в виде текста
Этап 5. Анализ тональности (сантимент анализ)

- Триадный минимум определения тональности: положительный, нейтральный или отрицательный тон сообщения.
- Многокритериальный анализ: вводится шкала тональности, например от -1 до +1.
- Более подробный анализ позволяет выявить детали тональности в различные категории (взволновано, шутливо, нейтрально, угрожающе, и т.д.).
- В простейшей реализации используются словари с заранее прописанными коэффициентами тональности для слов. Например, «люблю: +1; ненавижу: -1, пойду: 0».
- Широко используется известный словарь тональности русского языка - скачать датасет тонального словаря русского языка можно тут (примерно 28 тыс.слов): https://github.com/dkulagin/kartaslov/tree/master/dataset/emo_dict

13

Обработка ввода в виде текста
Этап 6. Классификация по темам

- Бывает бинарная классификация и мультиклассовая классификация.
- **Бинарная** позволяет определить – соответствует ли фраза (текст) нужной тематике или нет.
- **Мультиклассовая** определяет для каждого класса вероятность (близость) фразы(текста) к соответствующему классу, описывающему какую-то тему.
- Простейшая реализация – на основе словаря терминов, словосочетаний, слов привязанных к некоторым темам

14

Генерация ответа

1. Алгоритмы выбора заготовленных фраз и реплик.
2. Алгоритмы действия по сценарию (случайный или по контексту переход к рекламе).
3. Использование заранее подготовленных правил.
4. Автоматическая генерация реплики по контексту.
5. Синтез речи

15

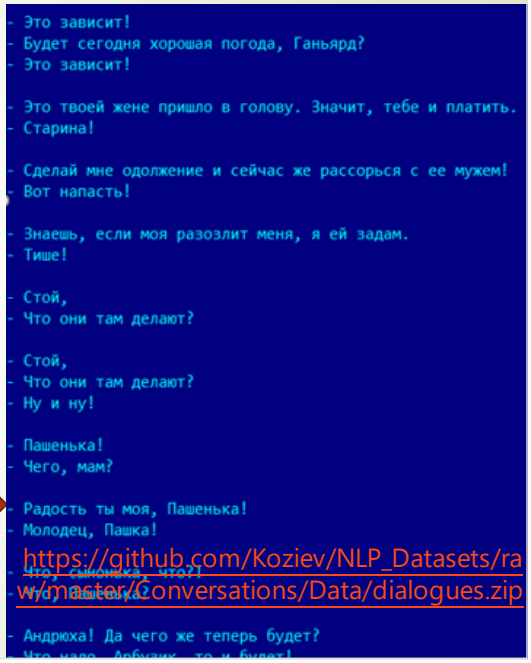
Расширение интеллектуальных возможностей чат-бота. Использование машинного обучения (ML) для анализа намерений. Основные этапы

1. Провести векторизацию фразы. Векторизация текста – это способ перевода текстов в числовые тензоры, или векторы чисел. Делается для того, чтобы подать эти векторы на вход нейронной сети – классификатора. Используется библиотека sklearn.
2. Создать классификатор на основе нейронной сети.
3. Обучить нейронную сеть.
4. Подать на нейронную сеть введенную фразу (предварительно тоже векторизованную).
5. Получить ответ нейронной сети – классификация намерения.

Использование стандартных диалогов из книг/интернета для поиска типовых ответов

16

- Можно настроить чат-бот так, чтобы он искал в датасете типовые вопросы и давал ответы. Для этого должен быть большой датасет, состоящий из пар строк: вопрос-ответ. Чат-бот ищет наиболее подходящий(близкий по смыслу) вопрос и выдает заготовленный ответ.
- Фрагмент из датасета диалогов.
- Исходный датасет нужно подготовить: убрать длинные диалоги, оставить только первые две строки:вопрос–ответ; почистить повторы вопросов.



https://github.com/Koziev/NLP_Datasets/raw/master/Conversations/Data/dialogues.zip

17

ЗАКЛЮЧЕНИЕ

- Рассмотрены основные этапы разработки чат-ботов, ведущих диалог с пользователем на разные темы, но имеющий определенную задачу – ненавязчиво информировать пользователя о характеристиках/достоинствах какого-то продукта/товара и по возможности предложить пользователю его покупку
- Приведены примеры интеллектуализации алгоритмов работы чат-ботов