МИНОБРНАУКИ РОССИИ

САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ «ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)

Кафедра Математического обеспечения и применения ЭВМ

ОТЧЕТ

по научно-исследовательской работе

Тема: Информационная система для тренировки публичных выступлений

Студентка гр. 6304 _____ Тарасова А.А.

Руководитель _____ Заславский М.М.

Санкт-Петербург

2021

ЗАДАНИЕ
НА НАУЧНО-ИССЛЕДОВАТЕЛЬСКУЮ РАБОТУ

Студентка Тарасова А. А.

Группа 6304

Тема научно-исследовательской работы: создание критерия оценки совпадения содержимого презентации и речи докладчика на основе ключевых слов

Содержание пояснительной записки:
«Содержание», «Глоссарий», «Постановка задачи», «Результаты работы в весеннем семестре», «План работы на весенний семестр», «Заключение», «Список использованных источников»

Предполагаемый объем пояснительной записки:
Не менее 10 страниц.

Дата выдачи задания: 20.11.2020

Дата сдачи отчета: 09.06.2021

Дата защиты отчета: 09.06.2021

Студентка ____ Тарасова А.А.

Руководитель ____ Заславский М.М.

ОГЛАВЛЕНИЕ

ГЛОССАРИЙ

NLP (Natural Language Processing) – обработка естественного языка – подраздел информатики и искусственного интеллекта, посвященный анализу естественных (человеческих) языков компьютером.

Токенизация – сегментация, разделение.

Лемматизация – приведение в начальную форму или унифицирование.

Стемминг – часть алгоритма распознавания слов, отвечающая за определение морфем. Используется для того, чтобы при обработке отбрасывать окончания и суффиксы, рассматривать только значимую часть слов – корень.

Метрика TF×IDF (*term frequency — inverse document frequency*) — частота употребления слова в документе в сравнении с частотой употребления слова в целом.

ПОСТАНОВКА ЗАДАЧИ

Основной целью публичного выступления является привлечение внимания публики к результатам работы или к некоторой проблеме. Важно удерживать внимание слушателя и предоставлять ему как можно больше возможностей для восприятия информации. С этой целью доклады зачастую сопровождаются презентацией.

Однако, чтобы презентация служила помощником, а не вызывала диссонанс у зрителей, содержимое слайдов должно соответствовать речи говорящего. Целью данной работы является добавление в существующую систему тренировки публичных выступлений критерия оценки такого соответствия. В качестве предмета исследования выступает сравнение ключевых слов речи докладчика и его презентации.

Составление такого критерия состоит из следующих этапов:

- Рассмотрение механизма извлечения ключевых слов и его реализация на языке Python
- Адаптация под извлечение КС для распознанной устной речи
- Создание механизма сравнения текстов по ключевым словам с учетом специфики решаемой задачи – расширения системы тренировки публичных выступлений
- Добавление критерия в существующую систему

К требованиям также можно отнести то, что тренажер публичных выступлений должен выдавать оценку качества за время, комфортное для пользователя (за несколько секунд), а значит, обработка критерия не должна занимать большее время, и необходимость использования opensource ПО, свободного от внешних зависимостей.

РЕЗУЛЬТАТЫ РАБОТЫ В ВЕСЕННЕМ СЕМЕСТРЕ

Введение

Поставленная задача относится к *NLP* – области, занимающейся обработкой естественных языков [1]. Спецификой задач данной области является зависимость от языка, работа с которым ведется.

Для извлечения ключевых слов особенности основными проблемами являются многозначность слов, зависимость от контекста, работа со словосочетаниями и попадание общеупотребимых слов в список ключевых.

Алгоритмы извлечения ключевых слов [2,3] призваны минимизировать влияние обозначенных проблем на точность распознавания, и исследования в данной области ведутся очень активно, так как ключевые слова – это механизм навигации в базе научных статей и основа поисковых запросов, а автоматизированное их извлечение позволяет упростить работу с этим механизмом.

Используемые технологии

В весеннем семестре было принято решение использовать библиотеку `nltk` и синтаксический анализатор `pymorphy2`. Они уже задействованы при написании тренажера публичных выступлений, а значит, это самый простой и надежный способ избежать дополнительных зависимостей.

`Nltk` [4] отвечает за *токенизацию* (в данном случае была выбрана токенизация на слова с использованием регулярных выражений), *стемминг* и за исключение так называемых стоп-слов. К ним относятся междометии, местоимения, слова-заполнители пауз и прочее. *Стемминг* представляет собой выделение значимой части слова (корня), избавление от приставок, суффиксов и окончаний. Он используется для минимизации последствий ошибок распознавания речи, что будет описано в разделе с описанием алгоритма. [1,5]

`Pymorphy2` [6] отвечает за морфологический анализ слов. В данной работе использована л*емматизация* – это приведение слова в начальную форму или унифицирование. [1] Иначе говоря, это способ воспринимать разные формы одного слова как одно слово. От части с этой задачей справляется и стемминг. И то, и другое работает не полностью верно с многозначными словами, но стемминг является более грубым механизмом, который будет сливать похожие слова с разными значениями в одно, в то время как лемматизация учитывает

контекст слова и работает более аккуратно. В основе pymorpy2 лежит словарь OpenCorpora, который постоянно расширяется.

Извлечение ключевых слов

Извлечение ключевых слов строится следующим образом:

1) Токенизация с использованием регулярных выражений nltk.tokenize.RegexpTokenizer

2) Исключение слов, входящих в nltk.stopwords и знаков пунктуации из string.punctuation

3) Лемматизация через pymorphy2.parse().normal_form

4) Подсчет tf

Term Frequency – частоты упоминания слова в тексте. Она очень проста по своей сути. Принимать ее в качестве единственной метрики не стоит из-за того, что общие слова, не имеющие стилистической окраски и не относящиеся к предметной области, могут иметь высокую частоту и, соответственно, попадать в список ключевых слов. Чтобы снизить вес таких слов, используется df.

5) Подсчет df

Document Frequency (DF) – частота употребления слова в корпусе документов. Она используется для снижения веса общеупотребительных слов. Иначе говоря, данная метрика учитывает, в каком количестве текстов из корпуса встречается данное слово, а значит, чем более узкой спецификой будет обладать слово, тем больший вес будет ему назначен

$$DF = \frac{dc(P, C)}{|C|}$$

где $dc(P, C)$ – количество документов в корпусе статей, в которые входит терм, а $|C|$ – число документов в этом корпусе.

6) Подсчет tf-idf

Term Frequency – Inverse Document Frequency) – произведение метрик TF и IDF (метрика, обратная DF), позволяющая выразить частоту употребления слов с

понижением веса общеупотребимых терминов. Именно эта метрика лежит в основе большинства алгоритмов выделения ключевых слов.

$$TF - IDF = \frac{TF}{DF} = TF \times IDF$$

Недостатком данной метрики является необходимость использования корпуса текстов, но в контексте решаемой задачи это не выступает проблемой – тексты выступлений хранятся в базе и могут быть использованы в расчетах. [7]

7) Выделение ключевых слов на основе tf-idf метрики либо по уровню (значение метрики нормируется, а затем задается уровень, выше которого слова будут восприняты как ключевые слова), либо по количеству (заданное количество слов с наибольшим значением)

Алгоритм сравнения на основе ключевых слов

В силу предположения о том, что на презентацию выносятся ключевые аспекты речи, наиболее важные мысли и план того, что должно быть произнесено, было принято решение придавать большую значимость ключевым словам в презентации, чем словам в устной речи.

Самый простой способ сравнения транскрипции устной речи с содержимым презентации – это нахождение процента совпадений отобранных ключевых слов. Он не учитывает разницу значимости слов в зависимости от того, произнесены они докладчиком или написаны на слайде. Еще один существенный недостаток обусловлен спецификой решаемой задачей: распознавание речи работает неидеально, зависит от дикции докладчика и совсем нетолерантно к англицизмам или узко специфицированным словам (которые вполне могут использоваться в публичных выступлениях). Таким образом, этот подход не решает поставленную задачу.

Было принято решение обрабатывать несовпадения с использованием стемминга. Суть алгоритма заключается в следующем: если слово является ключевым с точки зрения презентации, то проверяется его вхождение в произнесенный докладчиком текст. Если и там оно является ключевым, то считаем это удачным исходом и добавляем его к итоговой оценке (при этом вес зависит от части речи, принято полагать, что наибольший смысл содержится в существительных, далее следуют глаголы, а затем прилагательные, причастия, числительные и наречия, остальные же части речи имеют меньшую значимость). Если оно не вошло в список ключевых, считаем такой исход неудачным и понижаем оценку.

В случае, когда рассматриваемое слово вовсе не встречалось среди токенов речи докладчика, применяем стемминг. Значимая часть слова сравнивается с токенами речи (проверка вхождения строки в подстроку). Если вхождение найдено, определяем, было ли это слово частоупотребляемым и действуем аналогично сравнению без стемминга, принимая слово близким по значению. Если же вхождений не найдено, считаем, что слово распознано неправильно (не входит в словарь распознавателя) и дальнейшее его рассмотрение некорректно – такое слово не оказывает влияния на оценку выступления.

Результаты

Работа программы проверялась на текстовом файле, полученным после распознавания речи системой публичных выступлений. По нему была составлена презентация, примерно соответствующая содержанию речи, далее текст с нее был получен через pdf_parser, входящий в состав системы.

Так как отбор ключевых слов производится по уровню метрики tf-idf (проводится нормализация, поэтому на вход подается число в промежутке от 0 до 1), программа на одном и том же тексте была запущена при разных значениях.

Отметим, что в рассматриваемом файле текст распознан очень чисто, при прочтении не возникает смысловых несостыковок, он скорее всего был предварительно подправлен.

Tf-idf речи	Алгоритм со стеммингом	Алгоритм без стемминга
0.1	0.9156118143459916	0.9079497907949791
0.2	0.7468354430379747	0.7405857740585774
0.3	0.569620253164557	0.5648535564853557
0.4	0.48523206751054854	0.4811715481171548

Измерения проводились для фиксированного уровня метрики ключевых слов в презентации – 0.4. Тогда список ключевых слов выглядит следующим образом:

```
Ключевые в презентации {'точка', 'камера', 'метод', 'работа', 'трёхмерный', 'здание', '3d', 'sfm', 'задача', 'модель', 'программа', 'фотограмметрия', 'сравнение', 'особенность'}
```

Полученные ключевые слова

- Уровень 0.4:

```
Ключевые в речи {'модель', 'программа', 'трёхмерный', 'который', 'точка', 'снимок', 'особенность', 'здание', 'это', 'метод', 'объект'}
```

- Уровень 0.3:

```
Ключевые в речи {'точка', 'объект', 'среди', 'каждый', 'сравнение', 'особенность', 'модель', 'программа', 'также', 'представить', 'снижение', 'который', 'камера', 'трёхмерный', 'здание', 'это', 'sfm', 'получать', 'реконструкция', 'метод', 'использовать', 'снимок'}
```

- Уровень 0.2:

```
Ключевые в речи {'снимок', 'процесс', 'фотограмметрия', 'объект', 'это', 'точка', 'среди', 'метод', 'разный', 'сравнение', 'вид', 'sfm', 'слайд',
```

```
'снижение', 'реконструкция', 'метрика', 'трёхмерный', 'программа', 'камера',
'загрубление', 'также', 'иметь', 'получать', 'каждый', 'использовать',
'который', 'модель', 'особенность', 'представить', 'здание'}
```

- Уровень 0.1:

```
Ключевые в речи {'слайд', 'фотограмметрия', 'метрика', 'получать', 'среди',
'иметь', 'загрубление', 'осадки', 'процесс', 'влечь', 'однако', 'данные',
'набор', 'восстановление', 'сравнительный', 'sfm', 'камера', 'другой',
'сторона', 'разрешение', 'больший', 'облако', 'разный',
'среднеквадратический', 'снимок', 'получить', 'данный', 'метод',
'реконструкция', 'tls', 'отметить', 'погодный', 'вид', 'задача', 'создать',
'возникать', 'ошибка', 'сравнение', 'снижение', 'анализ', 'выявить',
'последовательность', 'тот', 'фотограмметрический', 'это', 'программа',
'необходимо', 'соответствующий', 'который', 'точка', 'особенность', 'модель',
'представить', 'каждый', 'два', 'фотографировать', 'использовать',
'последний', 'качество', 'трёхмерный', 'объект', 'здание', 'также'}
```

ЗАКЛЮЧЕНИЕ

Разобран общий алгоритм выделения ключевых слов. Рассмотрена работа аналогов используемых технологий.

За весенний семестр был написан скрипт для сравнения ключевых слов, полученных из текстов 2 файлов. Он частично адаптирован под поставленную задачу благодаря алгоритму обработки предположительно некорректно распознанных слов. В его состав входят расчет tf-idf метрики, лемматизация, стемминг, токенизация и сравнение текстов по ключевым словам. На вход принимаются файлы и уровни метрики tf-idf для каждого из текстов. При этом принято, что ключевые слова презентации имеют больший вес.

Данный скрипт в дальнейшем будет внедрен в систему тренировки публичных выступлений.

На данный момент он располагается в репозитории [8], куда скопированы и части системы тренировки публичных выступлений, задействованные для обработки аудиофайлов и распознавания текста. В следующем семестре он будет перенесен в репозиторий системы.

ПЛАН РАБОТЫ В ВЕСЕННЕМ СЕМЕСТРЕ

- Внедрение в систему тренировки публичных выступлений разработанного критерия, его адаптация для работы с сравнением по отдельным слайдам
- Тестирование корректности работы на реальных данных: распознанных текстах и загруженных презентациях, в том числе на «неудачных» примерах работы распознавателя, зашумленных файлах
- Установление оптимальных пороговых значений метрики tf-idf или количества ключевых слов для сравнения
- Отображение полученного значения критерия в оценку выступления с учетом, оценка необходимости полного вхождения ключевых слов презентации в речь докладчика
- Загрузка текстов в корпус из базы данных, используемой в существующей системе

СПИСОК ЛИТЕРАТУРЫ

1. Основы Natural Language Processing для текста // habr.com URL: https://habr.com/ru/company/Voximplant/blog/446738/

2. KEA - Applications of Ontology Engineering on MathematicalNatural Language Texts / S. Jeschk, N. Natho, M. Wilke // Center of Information Technologies (RUS), University of Stuttgart, MuLF, Berlin University of Technology, Germany, (IITS), University of Stuttgart, Germany

3. Е.В. Соколова, О.А. Митрофанова // Автоматическое извлечение ключевых слов и словосочетаний из русскоязычных текстов с помощью алгоритма KEA / Спб: изд-во СПбГУ, 2018

4. Описание библиотеки nltk // https://www.nltk.org/

5. Предобработка текста в NLP // python-school.ru URL: https://python-school.ru/nlp-text-preprocessing/

6. Описание синтаксического анализатора pymorphy2 // pymorphy2.readthedocs.io URL: https://pymorphy2.readthedocs.io/en/latest/

7. Статья по извлечению ключевых выражений // habr.com URL: https://habr.com/ru/post/468141/

8. Репозиторий с разработанным кодом // github.com URL: https://github.com/AATarasova/Keywords