МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ
ФГБОУВО «Пермский государственный национальный исследовательский университет»
Механико-математический факультет
Кафедра информационных технологий

Отчет по программе для автоматического составления тестов по тексту на естественном языке.

Работу выполнили студентки группы *ФИТ-17 нм 1* курса
механико-математического факультета
Ракина Валерия Денисовна
Булатова Дарья Андреевна

Пермь 2017

Предметная область:

Разрабатывается информационная система, которая позволит частично автоматизировать деятельность преподавателя по некоторой дисциплине. В данном примере ИС для дисциплины «Основы программирования». Информационная система должна содержать информацию для студентов указанной дисциплины, а именно лекционный материал, домашнее, практические и проверочные задания. Проверочные задания могут быть в виде теста. Необходимо предусмотреть возможность автоматического составления тестов.

Объект автоматизации:

Автоматизация деятельности преподавателя при составлении тестов.

Цель: написание программы для автоматического составления тестов.

Задачи:

- Структурировать исходный файл
- Сгенерировать вопросы и правильные ответы к ним
- Сохранить вопросы

Решение:

Поставленные задачи решались следующим образом.

- Для начала файл вручную был структурирован в формат json. (см. Рис. 1) Информация в файле представляется в виде коллокаций (денотат – отношение – денотат).
- Далее с помощью функции loadpairs денотатные пары были считаны из исходного файла. (см. Рис. 2)
- Затем из считанных денотатных пар были сгенерированы вопросы (см. Рис. 7) с помощью функции get_q_templs. (см. Рис. 3)
- На основе вопросов, составленных в нормальной разговорной форме (см. Рис. 4) была проведена работа по преобразованию окончаний слов

в вопросах, составленных программой, в нормальный вид (подставление правильного окончания, времени и числа) с помощью алгоритма N-грамм. (см. Рис. 5)

Основная программа, вызывающая все эти функции, имеет вид представленный на рисунке 6. (см. Рис. 6)


```
"denotatbase": [
    {
        "denotat2": "бытовая деятельность",
        "relation": "результат",
        "denotat1": "Твердые бытовые отходы"
    },
    {
        "denotat2": "гетерогенная смесь",
        "relation": "являться",
        "denotat1": "Твердые бытовые отходы"
    },
    {
        "denotat2": "компоненты",
        "relation": "иметь",
        "denotat1": "Твердые бытовые отходы"
    },
    {
        "denotat2": "бумага",
        "relation": "может быть",
        "denotat1": "компоненты"
    },
    {
        "denotat2": "пластмасса",
        "relation": "может быть",
        "denotat1": "компоненты"
    },
    {
        "denotat2": "пищевой отход",
        "relation": "может быть",
        "denotat1": "компоненты"
    },
```


Рис. 1

```
def loadpairs(filename = 'data.json'):
    mystem = Mystem()
    knowledge = {}
    try:
        kb = load(open(filename))
        for _ in kb.keys():
            pairs = kb[_]
            for pair in pairs:
                d1 = pair['denotat1']
                d2 = pair['denotat2']
                rl = pair['relation']
                lnk = ()
                for d in [d1, rl, d2]:
                    key = ''
                    val = []
                    prt = ''
                    # print(d)
                    for a in mystem.analyze(d):
                        try:
                            key += a['analysis'][0]['lex']
                            val += a['analysis']
                            prt += a['analysis'][0]['lex']
                        except KeyError:
                            key += a['text']
                            prt += a['text']
                        except IndexError:
                            key += a['text']
                            prt += a['text']
                    lnk += tuple([prt.strip('\n')])
                try:
                    knowledge[lnk] += 1
                except KeyError:
                    knowledge.update({lnk:1})
    except FileNotFoundError as fnfe:
        pass
    return knowledge
```

Рис. 2

```
def get_q_templs(
        znanie = {},
        kolvo = 5,
        form = "Правда ли что %s %s %s?"):
    rez = {}
    i = 1
    for d1, rel, d2 in znanie.keys():
        if i > kolvo: break
        if randint(0,10) > 2:
            if randint(0,1):
                rez.update({
                    form % (d2, rel, d1):{
                        "a":{"text":"да", "correct": 1},
                        "b":{"text":"нет", "correct": 0},
                        "c":{"text":"не знаю", "correct": 0},
                    }
                })
            else:
                for _d1, _rel, _d2 in znanie.keys():
                    if _d1 != d1 and rel != _rel:
                        rez.update({
                            form % (_d2, rel, d1):{
                                "a":{"text":"да", "correct": 0},
                                "b":{"text":"нет", "correct": 1},
                                "c":{"text":"не знаю", "correct": 0},
                            }
                        })
                        break
            i += 1
```

Рис. 3

```
Правда ли что гетерогенная смесь является твердым бытовым отходом?
Правда ли что бумага является твердым бытовым отходом?
Правда ли что бытовая деятельность может быть компонентом?
Правда ли что стекло может быть компонентом?
Правда ли что древесина может быть компонентом?
Правда ли что бытовая деятельность может быть компонентом?
Правда ли что резина может быть компонентом?
Правда ли что бытовая деятельность может быть фактором?
Правда ли что осень может быть сезоном?
Правда ли что бытовая деятельность результат твердых бытовых отходов?
Правда ли что бумага имеет твердые бытовые отходы?
Правда ли что пластмасса может быть компонентом?
Правда ли что пищевой отход может быть компонентом?
Правда ли что бытовая деятельность может быть компонентом?
Правда ли что стекло может быть компонентом?
Правда ли что кожа может быть компонентом?
Правда ли что бытовая деятельность может быть компонентом?
Правда ли что гетерогенная смесь является твердым бытовым отходом?
Правда ли что компонент имеет твердые бытовые отходы?
Правда ли что бытовая деятельность может быть компонентом?
Правда ли что бумага результат твердых бытовых отходов?
Правда ли что бумага является твердым бытовым отходом?
Правда ли что пластмасса может быть компонентом?
Правда ли что бытовая деятельность может быть компонентом?
```

Рис. 4

```
import re
from collections import defaultdict
from random import uniform

r_alphabet = re.compile(u'[а-яА-Я0-9-]+|[.,:;?!]+')
r_jo = re.compile('ё')

def train(corpus = 'voprosi'):
    lines = gen_lines(corpus)
    tokens = gen_tokens(lines)
    trigrams = gen_trigrams(tokens)

    bi, tri = defaultdict(lambda: 0.0), defaultdict(lambda: 0.0)

    for t0, t1, t2 in trigrams:
        bi[t0, t1] += 1
        tri[t0, t1, t2] += 1

    model = {}
    for (t0, t1, t2), freq in tri.items():
        if (t0, t1) in model:
            model[t0, t1].append((t2, freq/bi[t0, t1]))
        else:
            model[t0, t1] = [(t2, freq/bi[t0, t1])]
    return model

def gen_lines(corpus):
    data = open(corpus)
    for line in data:
        yield re.sub(r_jo, 'е', line.lower())

def gen_tokens(lines):
    for line in lines:
        for token in r_alphabet.findall(line):
            yield token

def gen_trigrams(tokens):
    t0, t1 = '$', '$'
    for t2 in tokens:
        yield t0, t1, t2
        if t2 in '.!?':
            yield t1, t2, '$'
            yield t2, '$','$'
            t0, t1 = '$', '$'
        else:
            t0, t1 = t1, t2
```

Рис. 5

```python
from loadpairs import loadpairs
from get_q_templs import get_q_templs
from gram_filter import gram_filter
from ngram import train
from json import dumps
from pymystem3 import Mystem
from generate_sentence import generate_sentence

ma = Mystem()

pairs = loadpairs(filename = 'data.json')
q_templs = get_q_templs(
    znanie = pairs,
    kolvo = 5,
    form = "Правда ли что %s %s %s?")
model = train(corpus = 'voprosi')
questions = gram_filter(q_templs,model)
kutokens = []
for question in questions:
    kutokens = []
    for a in ma.analyze(question):
        try:
            _ = a['analysis'][0]['lex']
            kutokens += [_]
        except (KeyError, IndexError):
            pass
    print(generate_sentence(model,kutokens))

```

Рис. 6


```
{'Правда ли что бытовой деятельность результат твердый
 бытовой отходы?': {'a': {'text': 'да', 'correct': 1},
 'b': {'text': 'нет', 'correct': 0}, 'c': {'text': 'не
 знаю', 'correct': 0}}, 'Правда ли что гетерогенный см
есь являться твердый бытовой отходы?': {'a': {'text':
'да', 'correct': 1}, 'b': {'text': 'нет', 'correct': 0
}, 'c': {'text': 'не знаю', 'correct': 0}}, 'Правда ли
 что компонент иметь твердый бытовой отходы?': {'a': {
'text': 'да', 'correct': 1}, 'b': {'text': 'нет', 'cor
rect': 0}, 'c': {'text': 'не знаю', 'correct': 0}}, 'П
равда ли что бумага мочь быть компонент?': {'a': {'tex
t': 'да', 'correct': 1}, 'b': {'text': 'нет', 'correct
': 0}, 'c': {'text': 'не знаю', 'correct': 0}}, 'Правд
а ли что бытовой деятельность мочь быть компонент?': {
'a': {'text': 'да', 'correct': 0}, 'b': {'text': 'нет'
, 'correct': 1}, 'c': {'text': 'не знаю', 'correct': 0
}}}
```


Рис. 7

Список литературы

- Н. М. Нестеров «Реферативный перевод: проблема смыслового свертывания и семантической адекватности»
- М.А. Павленко «Анализ методов решения задачи извлечения информации из текстов»
- Обзор существующих библиотек, решающих схожие задачи (Tesseract; nltk, python-mystem, pymystem3(лемматизация слова) для ЯП Python) и программ (mystem)

Ссылка на githab:
https://github.com/RakinaLera/ISforTests