МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ИМЕНИ Н. Э. БАУМАНА
Факультет информатики и систем управления
Кафедра теоретической информатики и компьютерных технологий

Лабораторная работа №7
по курсу «Автоматическая обработка текстов»

«Извлечение терминов из текста»

Выполнил:
студент группы ИУ9-11М
Беляев А. В.

Проверила:
Лукашевич Н. В.

Москва 2018

1 Цель работы

Изучить способы извлечения терминов из текста. Для этого необходимо взять текст учебника и извлечь из него конструкции вида *прилагательное существительное*. Извлеченные пары слов необходимо упорядочить, используя частоту вхождения пар и используя меру взаимной информации.

Затем необходимо сравнить полученные результаты.

2 Текст программы

```
import re
import math
import pymorphy2
from collections import defaultdict

morph = pymorphy2.MorphAnalyzer()

file_name = "cs-basics.txt"

TERMS_TO_OUTPUT = 20

TAG_ADJECTIVE = 'ADJF'
TAG_NOUN = 'NOUN'


def read_words(filename: str) -> list:
    contents = open(filename, 'r').read().casefold()
    clean = re.sub(r'[^a-z]', ' ', contents)   # add cyrillic to regex!
    splitted = re.compile(r'\s+').split(clean)
    return splitted[1:-1] # remove empty words - first and last

def print_terms(terms: list, filename: str):
    with open(filename, 'w+') as out:
        for term in terms[:TERMS_TO_OUTPUT]:
            print(f'{term[0][0]} {term[0][1]} -> {term[1]}', file=out)

def make_bigrams(words: list) -> list:
    bigrams = []
    i = 1
    while i < len(words):
        bigrams.append((words[i - 1], words[i]))
        i += 1
    return bigrams

def normalize_words(words: list) -> list:
    tags_to_remove = ['NPRO', 'PRED', 'PREP', 'CONJ', 'PRCL', 'INTJ']
    normalized = []
    for w in words:
        parsed = morph.parse(w)[0]
        # leave only words that matter - nouns, verbs, adjectives, etc.
        if parsed.tag.POS not in tags_to_remove:
            normalized.append(parsed.normal_form)
    return normalized
```

```
44
45
46 def find_terms(words: list) -> list:
47     is_term = lambda bi: bi[0].tag.POS == TAG_ADJECTIVE \
48                          and bi[1].tag.POS == TAG_NOUN
49
50     parsed_words = list(map(lambda w: morph.parse(w)[0], words))
51     parsed_bigrams = make_bigrams(parsed_words)
52
53     # find pairs of ADJECTIVE-NOUN among bigrams
54     potential_terms = list(filter(is_term, parsed_bigrams))
55     # leave original words
56     return list(map(lambda t: (t[0].word, t[1].word), potential_terms))
57
58
59 def count_mutual_info(term_count: dict, word_count: dict) -> list:
60     N = sum(map(lambda w: word_count.get(w), word_count))
61     term_measures = []
62     for term in term_count:
63         a = term[0]
64         b = term[1]
65         MI = math.log2(term_count[term]*N / (word_count[a]*word_count[b]))
66         term_measures.append((term, MI))
67     # sort by Mutual Information measure DESC
68     return sorted(term_measures, key=lambda term: term[1], reverse=True)
69
70
71 def main():
72     words = read_words(file_name)
73
74     normalized_words = normalize_words(words)
75     word_count = defaultdict(int)
76     for word in normalized_words:
77         word_count[word] += 1
78
79     normalized_terms = find_terms(normalized_words)
80     term_count = defaultdict(int)
81     for term in normalized_terms:
82         term_count[term] += 1
83
84     # get top terms by frequency
85     terms_by_frequency = sorted(term_count.items(), key=lambda term: term
           [1], reverse=True)
86     print_terms(terms_by_frequency, 'out_freq.txt')
87
88     # get top terms by mutual information measure
89     terms_by_mutual_info = count_mutual_info(term_count, word_count)
90     print_terms(terms_by_mutual_info, 'out_mi.txt')
91
92 if __name__ == '__main__':
93     main()
```

Листинг 1: Исходный код программы

3 Результаты тестирования

В качестве текста для исследования был выбран учебник «Основы информатики: Учебник для вузов» (автор – Малинина Л. А.). Из текста были убраны «лишние» части речи, а оставшиеся слова были нормализованы.

В результате тестирования были получены по 20 словосочетаний, представленных в Таблицах 1 и 2, упорядоченных по частоте использования и по мере взаимной информации соответственно. Словосочетания были выделены **жирным текстом**.

Можно сделать следующие выводы:

- У словосочетаний имеется устойчивый оборот, в котором они упоминаются. При этом значение словосочетания незначительно отличается от значения двух слов (например, «операционная система», «программное обеспечение»). Либо, значение может быть выведено, как в словосочетании «унарное умножение», «аналоговый преобразователь», «беглый взгляд». Либо, это может быть неделимый оборот, значение которого совершенно не выводимо из входящих в него слов: «всемирная паутина», «троянский конь», «нервная ткань».

- Одно из слов, входящих в словосочетание не употребляется без другого. Так, слово «троянский» практически не упоминается без слова «конь».

- Слова в терминах практически не поддаются замене: «электронная (электрическая? цифровая?) почта», «текстовый редактор (монтажер? преобразователь?)»

- Частотный список и Мера взаимной информации показали схожие невысокие результаты. Частотный список при этом содержит больше слов, «подходящих по контексту», т.е. относящихся к компьютерной терминологии. В то время, как мера взаимной информации сильно завышает значение для слов, употребленных единожды в тексте («профсоюзный взнос», «ежемесячная премия»)

- Некоторые слова были некорректно нормализованы или же являлись, в некотором роде, служебными и, таким образом, словосочетания не имеют смысла: «слоновый костя», «который мочь», «такой образ».

- Часть слов в словосочетаниях может быть заменена или дополнена синонимами: «справочная (поисковая, консультационная) система», «профсоюзный взнос (плата)», «конфиденциальная (закрытая, доверительная) информация (данные)». Слова полностью подчинены правилам.

- Значения словосочетаний «левая кнопка», «двойной щелчок» и других полностью складываются из значения обоих слов. И замены «левая» $\rightarrow$ «правая» или «двойной» $\rightarrow$ «тройной» не меняют общей сути: «какая-то кнопка», «несколько легких соударений чем-либо».

4 Выводы

В ходе работы были изучены методы извлечения терминов из текста и была произведена оценка полученных результатов.

Таблица 1: Частотный список, 8/20

операционный система
контекстный меню
диалоговый окно
компьютерный сеть
текстовый редактор
левый кнопка
информационный модель
программный обеспечение
локальный сеть
такой образ
предварительный просмотр
компьютерный моделирование
почтовый программа
левый панель
электронный почта
конфиденциальный информация
справочный система
двойной щелчок
который мочь
всемирный паутина

Таблица 2: Мера взаимной информации (MI), 9/20

аналоговый преобразователь
стеклянный пластина
троянский конь
беглый взгляд
пишущий машинка
нервный ткань
прибыльный инвестиция
визитный карточка
унарный умножение
ежемесячный премия
профсоюзный взнос
неоднозначный толкование
критический порог
испытательный стенд
зимний месяц
гигантский хранилище
атомный энергия
слоновый костя
сходный тираж
читательский спрос