Министерство Науки и Высшего Образования Российской Федерации

Национальный Исследовательский Институт Высшая Школа Экономики

Факультет Компьютерных Наук

Школа Анализа Данных и Искусственного Интеллекта

ДОМАШНЕЕ ЗАДАНИЕ №1

СЕГМЕНТАЦИЯ И МОРФОЛОГИЧЕСКИЙ АНАЛИЗ ТЕКСТА, СТАТИСТИКА

Компьютерная лингвистика и анализ текстов

Студент *М.Д. Кирдин*

Преподаватель *Е.И. Большакова*

Москва, 2025г.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

В данной работе была поставлена цель провести исследование качества автоматического извлечения коллокаций с использованием различных мер ассоциации, при помощи программы на языке *Python*.

Для токенизации, лемматизации, теггинга и синтаксического парсинга был использован модуль *spaCy*. Для оценки был использован датасет *russian_cleared_wikipedia*, являющийся набором статей на различные темы из русскоязычной Википедии.

1 ХОД РАБОТЫ

В данной работе было проведено извлечение коллокаций типов «A N» и «N N». Благодаря интерфейсу предлагаемому модулем *spaCy*, достаточно было найти существительные, родителями которых являлись прилагательные или существительные.

Было решено использовать следующие меры:

- $Dice = \dfrac{2f(ab)}{f(a) + f(b)}$,
- $MI = \log_2 \left(\dfrac{f(ab) \cdot N}{f(a) \cdot f(b)} \right)$,
- $MI^3 = \log_2 \left(\dfrac{f^3(ab) \cdot N}{f(a) \cdot f(b)} \right)$,
- $\text{T-score} = \dfrac{f(ab) - \dfrac{f(a)f(b)}{N}}{\sqrt{f(ab)}}$,

где $f(a), f(b)$ – абсолютные частоты употребления элементов словосочетания, $f(ab)$ – абсолютная частота употребления словосочетания и N – общее число лемм/словоформ в тексте, т.к. статистики для лемм и словоформ рассмотрены раздельно.

С результатами работы программы можно ознакомиться в таблицах 1-4.

Таблица 1. Возможные коллокации при использовании меры *Dice*

Позиция	Пара лемм	Значение меры	Пара словофром	Значение меры
# 1	стыдились» наги	1.000000	стыдились» наги	1.000000
# 2	вальковатый шишками	1.000000	вальковатыми шишками	1.000000
# 3	судорожный припадки	1.000000	декоративном садоводстве	1.000000
# 4	сопровождаемому дрожью	1.000000	непроходима густа	1.000000
# 5	абсентная эпилепсия	1.000000	новогодней ёлки	1.000000
# 6	субальпийский криволесье	1.000000	ноевом ковчеге	1.000000
# 7	абазгийская архиепископия	1.000000	растительным орнаментам	1.000000
# 8	герцинской складчатость	1.000000	волнистым узорам	1.000000
# 9	сбрасывая рассола	1.000000	параноидной шизофрении	1.000000
# 10	чехословацкую республику.16	1.000000	сомнительных забегаловках	1.000000
# 11	сформированным профсоюзами	1.000000	недобросовестными производителями	1.000000
# 12	назначаемыми рейхсканцлер	1.000000	судорожные припадки	1.000000
# 13	передвижной медпункт	1.000000	сопровождаемому дрожью	1.000000
# 14	бессальниковыми мешалками	1.000000	абсентная эпилепсия	1.000000
# 15	апериодический ритмической	1.000000	платиновой горелки	1.000000
# 16	никелированный чайник	1.000000	субальпийское криволесье	1.000000
# 17	еобъемлющую идеологизация	1.000000	кавказский тетерев	1.000000

Позиция	Пара лемм	Значение меры	Пара словофром	Значение меры
# 18	гусаревский овцесовхоз	1.000000	абазгийская архиепископия	1.000000
# 19	низвергнутые титаны	1.000000	абхазскими литераторами	1.000000
# 20	ежовый рукавица	1.000000	покупательские авансы	1.000000

Таблица 2. Возможные коллокации при использовании меры MI

Позиция	Пара лемм	Значение меры	Пара словофром	Значение меры
# 1	стыдились» наги	18.992684	стыдились» наги	18.992684
# 2	вальковатый шишками	18.992684	вальковатый шишками	18.992684
# 3	сопровождаемому дрожью	18.992684	непроходима густа	18.992684
# 4	абсентная эпилепсия	18.992684	новогодней ёлки	18.992684
# 5	субальпийский криволесье	18.992684	растительным орнаментам	18.992684
# 6	абазгийская архиепископия	18.992684	волнистым узорам	18.992684
# 7	герцинской складчатость	18.992684	параноидной шизофрении	18.992684
# 8	сбрасывая рассола	18.992684	сомнительных забегаловках	18.992684
# 9	чехословацкую республику.16	18.992684	недобросовестными производителями:	18.992684
# 10	сформированным профсоюзами	18.992684	сопровождаемому дрожью	18.992684
# 11	назначаемыми рейхсканцлер	18.992684	абсентная эпилепсия	18.992684
# 12	передвижной медпункт	18.992684	платиновой горелки	18.992684
# 13	бессальниковыми мешалками	18.992684	субальпийское криволесье	18.992684

Позиция	Пара лемм	Значение меры	Пара словофром	Значение меры
# 14	апериодический ритмической	18.992684	кавказский тетерев	18.992684
# 15	никелированный чайник	18.992684	абазгийская архиепископия	18.992684
# 16	еобъемлющую идеологизация	18.992684	абхазскими литераторами	18.992684
# 17	гусаревский овцесовхоз	18.992684	покупательские авансы	18.992684
# 18	низвергнутые титаны	18.992684	стартовые ускорители	18.992684
# 19	ежовый рукавица	18.992684	подводными лодками	18.992684
# 20	неприличие оппортунизм	18.992684	летательными аппаратами	18.992684

Таблица 3. Возможные коллокации при использовании меры MI^3

Позиция	Пара лемм	Значение меры	Пара словофром	Значение меры
# 1	населённый пункт	24.725356	xix века	23.647912
# 2	xix век	24.516431	населённые пункты	23.410258
# 3	учебный заведение	23.840716	xx века	23.311953
# 4	xx век	23.799354	мировой войны	23.091003
# 5	железный дорога	23.317423	известные носители	22.861682
# 6	мировой война	23.258495	водного реестра	22.849124
# 7	xviii век	22.774249	второй половине	22.786880
# 8	сельский хозяйство	22.731393	культурного наследия	22.667941
# 9	заработный плата	22.396550	сельское хозяйство	22.657292
# 10	второй половина	22.283336	населённых пунктов	22.586651
# 11	xvii век	22.205239	железной дороги	22.467348
# 12	вооружённый сила	21.832522	стальных труб	22.458017
# 13	ванный комната	21.788099	российской империи	22.345181
# 14	летательный аппарат	21.657592	заработная плата	22.2010544
# 15	муниципальный образование	21.589014	железная дорога	22.033326

Позиция	Пара лемм	Значение меры	Пара словофром	Значение меры
# 16	xvi век	21.569048	географическая характеристика	22.026254
# 17	православный церковь	21.567829	натуральных чисел	22.012446
# 18	полётный палуба	21.543636	одномандатным округам	21.992684
# 19	слизистый оболочка	21.508105	сельском хозяйстве	21.922896
# 20	отечественный война	21.4979875	бассейновому округу	21.824084

Таблица 4. Возможные коллокации при использовании меры T-score

Позиция	Пара лемм	Значение меры	Пара словофром	Значение меры
# 1	xix век	17.784836	xix века	14.777780
# 2	xx век	15.399136	xx века	13.374665
# 3	xviii век	13.023393	мировой войны	10.614954
# 4	мировой война	12.733342	xviii века	9.806472
# 5	2010 год	12.123090	российской империи	9.679281
# 6	больший часть	11.952826	второй половине	9.210340
# 7	xvii век	11.825521	2011 года	9.134809
# 8	2011 год	11.421811	2010 года	8.847549
# 9	второй половина	10.833149	второй войны	8.681058
# 10	2009 год	10.678807	большая часть	8.288353
# 11	xvi век	10.632479	главным образом	8.235930
# 12	населённый пункт	10.387935	населённый пункт	10.387935
# 13	российский империя	10.302256	российский империя	10.302256
# 14	железный дорога	10.138591	железный дорога	10.138591
# 15	2014 год	9.875496	2014 год	9.8754966
# 16	2015 год	9.585918	2015 год	9.585918
# 17	второй война	9.547817	второй война	9.547817
# 18	2008 год	9.511432	2008 год	9.511432
# 19	xiii век	9.499602	xiii век	9.499602
# 20	2013 год	9.363738	2013 год	9.363738

РЕЗУЛЬТАТЫ

Вследствие того как заданы метрики, по таблицам 1 и 2 можно увидеть, что все коллокации из списка имеют одинаковое значение мер для $Dice$ и MI. Это объясняется тем, что пары слов/лемм, употребленные вместе один раз и ни разу не употребленные раздельно получают максимально возможное значение меры. Действительно – большинство попавших в эти списки коллокаций – узконаправленные термины, однако стоит заметить что в оба списка также попала устойчивая фраза «ежовые рукавицы». Также в список попали неверно токенизированные пары словоформ: «чехословацкую республику.16», «стыдились» наги».

Описанная проблема отсутствует в случаях мер MI^3 и T-score, т.к. в них количество взаимных взаимных употреблений имеет больший вес и его рост увеличивает значение метрики, а не уменьшает, что легко заметить по таблицам 3 и 4. Стоит заметить, однако, что обе меры часто воспринимают даты или временные промежутки как коллокации: «xix век», «2015 год» и т.д.

ЗАКЛЮЧЕНИЕ

Метрики $Dice$ и MI в силу специфики данных и ошибок токенизации проявили себя хуже всего в выделении коллокация как по леммам, так и по словоформам. Метрики T-score и MI^3 проявили себя лучше, при этом выделение колллокаций по леммам оказалось менее верным нежели чем по словоформам, что можно заметить по большей пропорции дат и временных промежутков в таблицах 3 и 4 для лемм. Наконец, мера MI^3 справляется с задачей извлечения коллокаций лучше остальных, особенно при подсчете статистик на словоформах.

ПРИЛОЖЕНИЕ А

Листинг А.1. Программа run.py

```
import spacy
from collections import defaultdict, Counter
from json import loads
from typing import Iterable, Callable
from spacy.symbols import amod, ADJ, NOUN
from numpy import log2, power, sqrt


collocation_types = [
        ("NOUN", "NOUN"),
        ("ADJ", "NOUN"),
        ("VERB", "NOUN"),
        ]


# Note that Dice accepts N only to have consistent signature with
    the other
# metrics. Although this defeats the whole purpose of the metric,
    I could not
# think of a better way to do this.
collocation_metrics = {
        "Dice": lambda f_ab, f_a, f_b, N: 2.0 * f_ab / (f_a + f_b
           ),
        "MI": lambda f_ab, f_a, f_b, N: log2(f_ab * N / f_a / f_b
           ),
        "MI3": lambda f_ab, f_a, f_b, N: log2(power(f_ab, 3) * N
           / f_a /f_b),
        "T-score": lambda f_ab, f_a, f_b, N: (f_ab - f_a * f_b /
           N) / sqrt(f_ab),
        }


def is_noun_noun_collocation(token):
    return (token.pos == NOUN) and (token.dep == amod) and (token
       .head.pos == NOUN) and (token.text != "-") and (token.head
       .text != "-")
```

```
def is_adj_noun_collocation(token):
    return (token.pos == ADJ) and (token.head.pos == NOUN) and (
       token.text != "-") and (token.head.text != "-")


def evaluate_metrics(collocation_candidates: dict, frequencies:
   dict, metrics: dict[str, Callable]=None) -> dict:
    """
    Evaluate provided metrics (or mutual information metric if
       they are not provided) for a given set of co-occuring word
        pairs and wordform/lemma frequencies.
    """
    N = sum(frequencies.values())
    for candidate in collocation_candidates:
        try:
            first_member, second_member = candidate.split(" ",
               maxsplit=2)
        except ValueError:
            print(f"Warning: !!! could not split {candidate} !!!"
               )

        f_ab = collocation_candidates[candidate]["n_uses"]
        f_a = frequencies[first_member]
        f_b = frequencies[second_member]

        # metrics are not evaluated for defective collocations
        if f_a == 0 or f_b == 0:
            print(f"Warning: failed to process '{first_member
               }'+'{second_member}'")
            continue

        if metrics is None:
            collocation_candidates[candidate]["MI"] = log2(f_ab *
                N / f_a / f_b)
        else:
            for metric in metrics:
                collocation_candidates[candidate][metric] =
                   metrics[metric](f_ab, f_a, f_b, N)
    return collocation_candidates
```

```
if __name__=="__main__":

    wordforms = []
    lemmi = []
    wordform_collocations = defaultdict()
    lemmi_collocations = defaultdict()

    nlp = spacy.load("ru_core_news_sm")

    with open("./wiki_dataset.json", "rb") as file:
        n_lines = sum(1 for _ in file)

    with open("./wiki_dataset.json", "r", encoding="ascii") as
       file:

        lines_read = 0
        for line in file:

            lines_read += 1
            line = line.strip()
            tokenized_line = nlp(loads(line)["sample"])
            if lines_read % 5 == 0 or lines_read == 1:
                print(f"Reading line {lines_read} / {n_lines}",
                   end='\r')

            for token in tokenized_line:

                if not (token.is_stop or token.is_punct):

                    wordforms += [token.text.lower()]

                    lemmi += [token.lemma_.lower()]

                    if is_noun_noun_collocation(token):
                        collocation_wordform = f"{token.text.
                           lower()} {token.head.text.lower()}"
                        collocation_lemma = f"{token.lemma_.lower
```

```
                    ()} {token.head.lemma_.lower()}"
                if not token.head in tokenized_line:
                    print(f"{collocation_lemma}\n{
                        collocation_wordform}")
                if collocation_wordform in
                    wordform_collocations:
                    wordform_collocations[
                        collocation_wordform]["n_uses"] +=
                         1
                else:
                    wordform_collocations[
                        collocation_wordform] = {"n_uses":
                         1, "type": "N N"}
                if collocation_lemma in
                    lemmi_collocations:
                    lemmi_collocations[collocation_lemma
                        ]["n_uses"] += 1
                else:
                    lemmi_collocations[collocation_lemma]
                        = {"n_uses": 1, "type": "N N"}

            if is_adj_noun_collocation(token):
                collocation_wordform = f"{token.text.
                    lower()} {token.head.text.lower()}"
                collocation_lemma = f"{token.lemma_.lower
                    ()} {token.head.lemma_.lower()}"
                if not token.head in tokenized_line:
                    print(f"{collocation_lemma}\n{
                        collocation_wordform}")
                if collocation_wordform in
                    wordform_collocations:
                    wordform_collocations[
                        collocation_wordform]["n_uses"] +=
                         1
                else:
                    wordform_collocations[
                        collocation_wordform] = {"n_uses":
                         1, "type": "A N"}
                if collocation_lemma in
                    lemmi_collocations:
```

```
                        lemmi_collocations[collocation_lemma
                            ]["n_uses"] += 1
                    else:
                        lemmi_collocations[collocation_lemma]
                            = {"n_uses": 1, "type": "A N"}

        if lines_read == 1000:
            break

lemmi_counts = Counter(lemmi)
wordform_counts = Counter(wordforms)

print("For lemmi:")
lemmi_collocations = evaluate_metrics(lemmi_collocations,
    lemmi_counts, collocation_metrics)

print("For wordforms:")
wordform_collocations = evaluate_metrics(
    wordform_collocations, wordform_counts,
    collocation_metrics)

def get_key(x:tuple, metric:str):
    if metric in x[1]:
        return x[1][metric]
    else:
        return -1e-6

n_colloc = 20

with open(f"./result_{n_colloc}.txt", "w", encoding="utf8")
    as file:

    for metric_name in collocation_metrics:
        out = f"топ-{n_colloc} словосочетаний(по леммам, метр
            ика {metric_name}):\n\n"
        file.write(out)
        for collocation in sorted(lemmi_collocations.items(),
            key = lambda x: get_key(x, metric_name), reverse=
            True)[:n_colloc]:
            out = f"{collocation[0]}:{collocation[1][
```

```
                metric_name]:0.6f}\n"
            file.write(out)

        out = "
            ==========================================================\
            n\n"
        file.write(out)

        out = f"топ-{n_colloc} словосочетаний(по словоформам,
            метрика {metric_name}):\n\n"
        file.write(out)
        for collocation in sorted(wordform_collocations.items
            (), key = lambda x: get_key(x, metric_name),
            reverse=True)[:n_colloc]:
            out = f"{collocation[0]}:{collocation[1][
                metric_name]:0.6f}\n"
            file.write(out)

        out = "
            ==========================================================\
            n\n"
        file.write(out)
```