



# Информационные ресурсы в финансовом мониторинге

---

НИЯУ МИФИ, КАФЕДРА ФИНАНСОВОГО МОНИТОРИНГА

КУРС ЛЕКЦИЙ

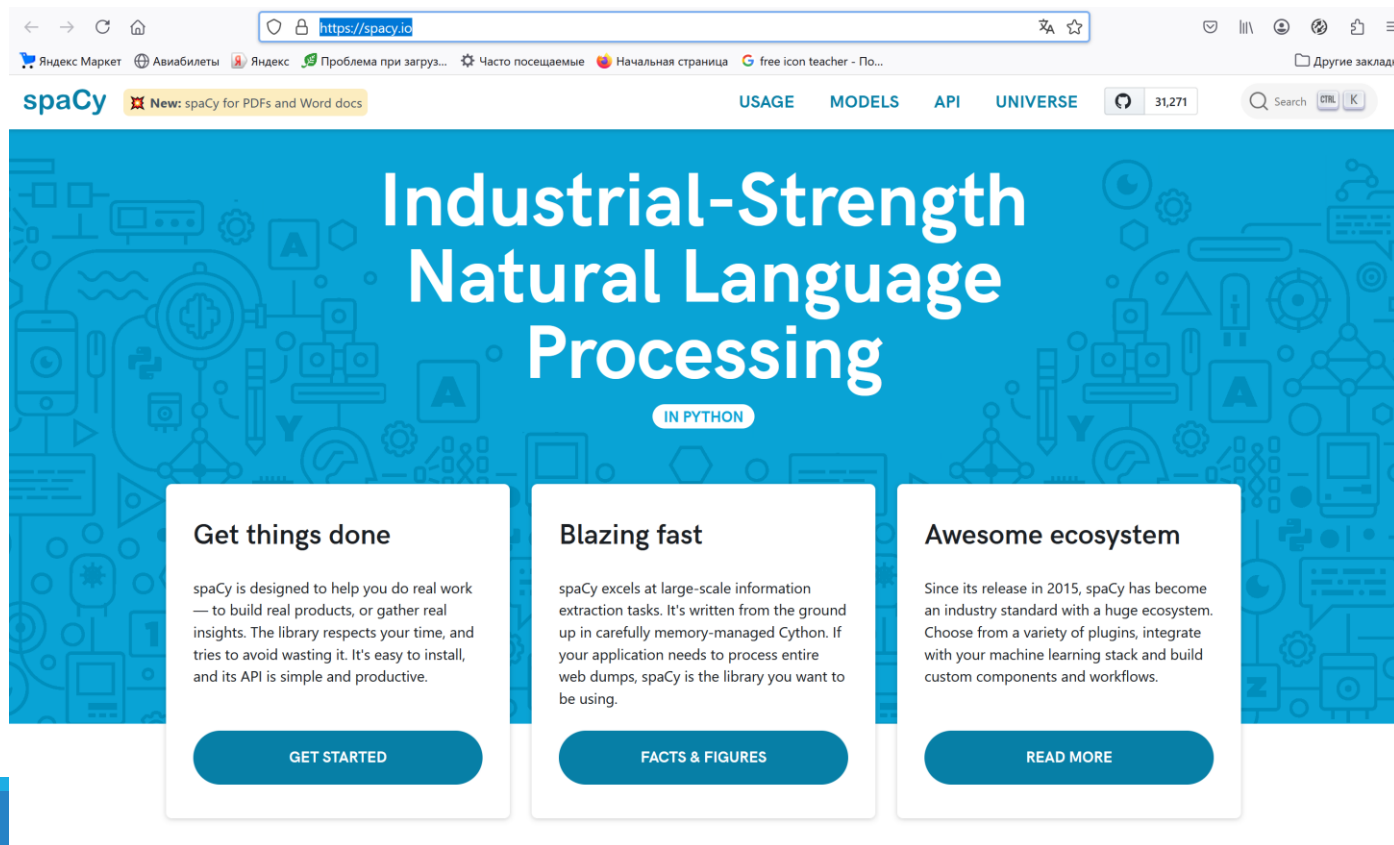
В.Ю. РАДЫГИН. ЛЕКЦИЯ 7

# Работа с русским языком

---

# Библиотека spaCy

Для работы с русским языком мы будем использовать другую библиотеку, так как корпус русского языка в NLTK не очень полный. В частности, мы рассмотрим библиотеку spaCy [1].



# Установка spaCy

---

Для установки spaCy установим саму библиотеку:

```
python -m pip install spacy
```

И модуль для русского языка [2]. Модуль для русского языка доступен в конфигурации small, medium и large. В данных примерах мы будем использовать минимальную конфигурацию:

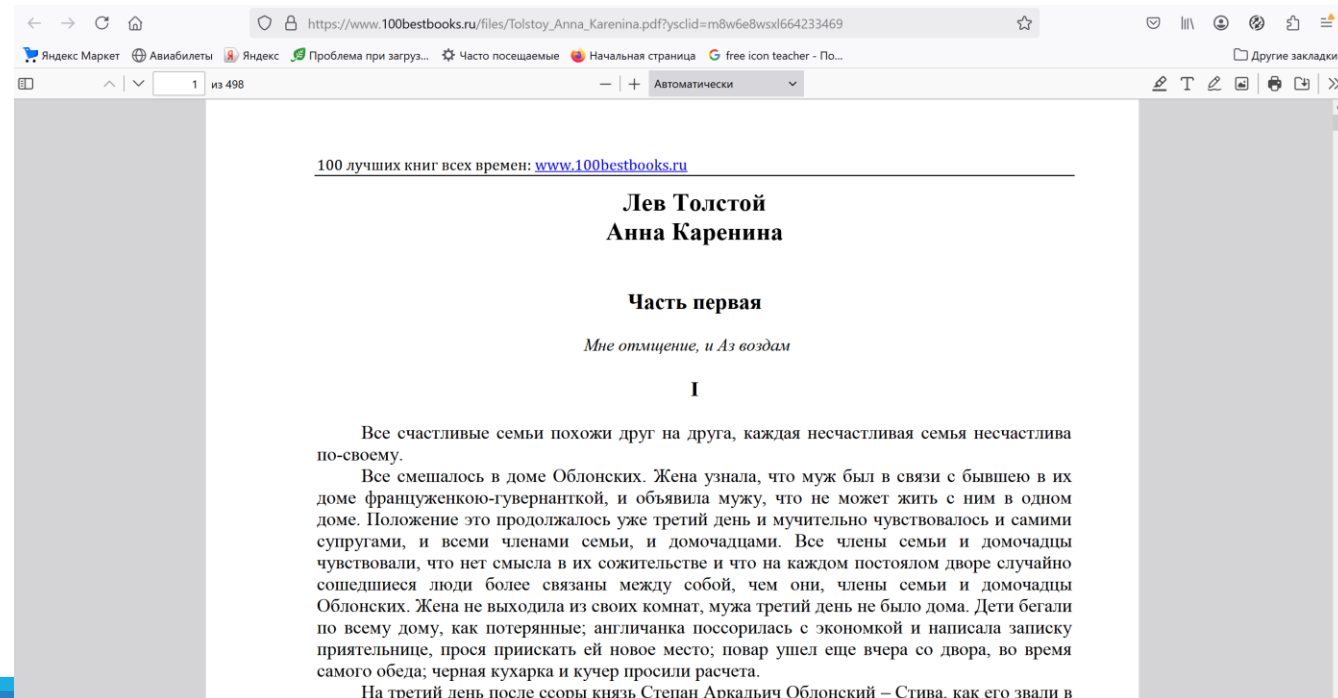
```
python -m spacy download ru_core_news_sm
```

Для более эффективного использования spaCy лучше использовать конфигурационную страницу установки: <https://spacy.io/usage>. На данной странице Вы сможете выбрать тип операционной системы, вид вычислительного устройства (GPU, если видеокарта хорошая или CPU иначе), языковой пакет и т.д.

# Задача для примеров

Построим дискретную функцию зависимости числа неповторяющихся слов (словаря) от общего числа слов текста для книги Анна Каренина:

[https://www.100bestbooks.ru/files/Tolstoy\\_Anna\\_Karenina.pdf?ysclid=m8w6e8wsxl664233469](https://www.100bestbooks.ru/files/Tolstoy_Anna_Karenina.pdf?ysclid=m8w6e8wsxl664233469)



# Шаг 1. Загрузка файла

---

```
*spacy_usage.py - G:\Works\Victor\Students\infres\2024\Lecture7\spacy_usage.py (3.12.4)*
File Edit Format Run Options Window Help
from urllib import request
from urllib.request import Request, urlopen

url = 'https://www.100bestbooks.ru/files/Tolstoy_Anna_Karenina.pdf?ysclid=m8w6e8wsxl664233469'

req = Request(url, headers={"User-Agent": "Mozilla/5.0"})
file = urlopen(req) # , context = ctx)
data = file.read()
file2 = open('ak.pdf', 'wb+')
file2.write(data)
file2.close()

file.close()
```

Ln: 15 Col: 22

# Шаг 1. Загрузка файла (текстом)

---

```
from urllib import request
from urllib.request import Request, urlopen

url = 'https://www.100bestbooks.ru/files/Tolstoy_Anna_Karenina.pdf?ysclid=m8w6e8wsxl664233469'

req = Request(url, headers={"User-Agent": "Mozilla/5.0"})
file = urlopen(req) # , context = ctx)
data = file.read()
file2 = open('ak.pdf', 'wb+')
file2.write(data)
file2.close()

file.close()
```

## Шаг 2. Выделение текста

```
*spacy_usage.py - G:\Works\Victor\Students\infres\2024\Lecture7\spacy_usage.py (3.12.4)*
File Edit Format Run Options Window Help

from io import StringIO
from pdfminer.pdfinterp import PDFResourceManager, PDFPageInterpreter
from pdfminer.converter import TextConverter
from pdfminer.layout import LAParams
from pdfminer.pdfpage import PDFPage
import os
import sys, getopt

def convert(fname, pages=None):
    if not pages:
        pagenums = set()
    else:
        pagenums = set(pages)

    output = StringIO()
    manager = PDFResourceManager()
    converter = TextConverter(manager, output, laparams=LAParams())
    interpreter = PDFPageInterpreter(manager, converter)

    infile = open(fname, 'rb')
    for page in PDFPage.get_pages(infile, pagenums):
        interpreter.process_page(page)
    infile.close()
    converter.close()
    text = output.getvalue()
    output.close()
    return text

text = convert("ak.pdf", pages = range(0, 497))
print(text[0:10000])
```



# Шаг 2. Выделение текста (текстом)

---

```
from io import StringIO

from pdfminer.pdfinterp import PDFResourceManager, PDFPageInterpreter

from pdfminer.converter import TextConverter

from pdfminer.layout import LAParams

from pdfminer.pdfpage import PDFPage

import os

import sys, getopt


def convert(fname, pages=None):

    if not pages:

        pagenums = set()

    else:

        pagenums = set(pages)
```

```
    output = StringIO()

    manager = PDFResourceManager()

    converter = TextConverter(manager, output, laparams=LAParams())

    interpreter = PDFPageInterpreter(manager, converter)

    infile = open(fname, 'rb')

    for page in PDFPage.get_pages(infile, pagenums):

        interpreter.process_page(page)

    infile.close()

    converter.close()

    text = output.getvalue()

    output.close()

    return text

text = convert("ak.pdf ", pages = range(0, 497))

print(text[0:10000])
```

# Результат



```
===== RESTART: G:\Works\Victor\Students\infres\2024\Iection7\spacy_usage.py =====
100 лучших книг всех времен: www.100bestbooks.ru

Лев Толстой

Анна Каренина

Часть первая

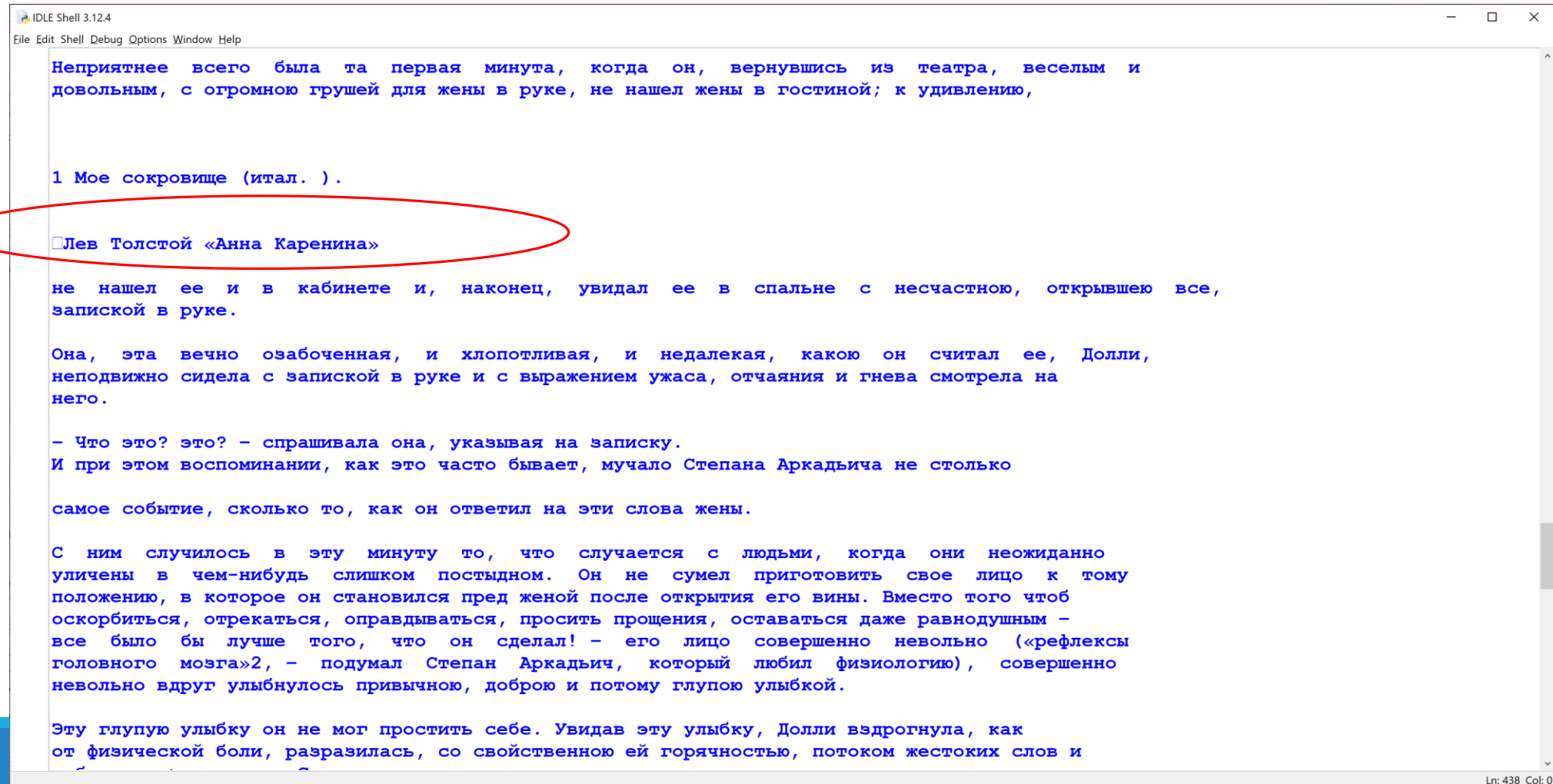
Мне отмщение, и Аз воздам

I

Все счастливые семьи похожи друг на друга, каждая несчастливая семья несчастлива
по-своему.

Все смешалось в доме Облонских. Жена узнала, что муж был в связи с бывшею в их
доме француженкою-гувернанткой, и объявила мужу, что не может жить с ним в одном
доме. Положение это продолжалось уже третий день и мучительно чувствовалось и самими
супругами, и всеми членами семьи, и домочадцами. Все члены семьи и домочадцы
чувствовали, что нет смысла в их сожительстве и что на каждом постоянном дворе случайно
сошедшиеся люди более связаны между собой, чем они, члены семьи и домочадцы
Облонских. Жена не выходила из своих комнат, мужа третий день не было дома. Дети бегали
```

# Результат



```
File Edit Shell Debug Options Window Help
Неприятнее всего была та первая минута, когда он, вернувшись из театра, веселым и
довольным, с огромною грушей для жены в руке, не нашел жены в гостиной; к удивлению,

1 Мое сокровище (итал. ).

Лев Толстой «Анна Каренина»

не нашел ее и в кабинете и, наконец, увидел ее в спальне с несчастною, открывшею все,
запиской в руке.

Она, эта вечно озабоченная, и хлопотливая, и недалекая, какою он считал ее, Долли,
неподвижно сидела с запиской в руке и с выражением ужаса, отчаяния и гнева смотрела на
него.

– Что это? это? – спрашивала она, указывая на записку.
И при этом воспоминании, как это часто бывает, мучало Степана Аркадьича не столько
самое событие, сколько то, как он ответил на эти слова жены.

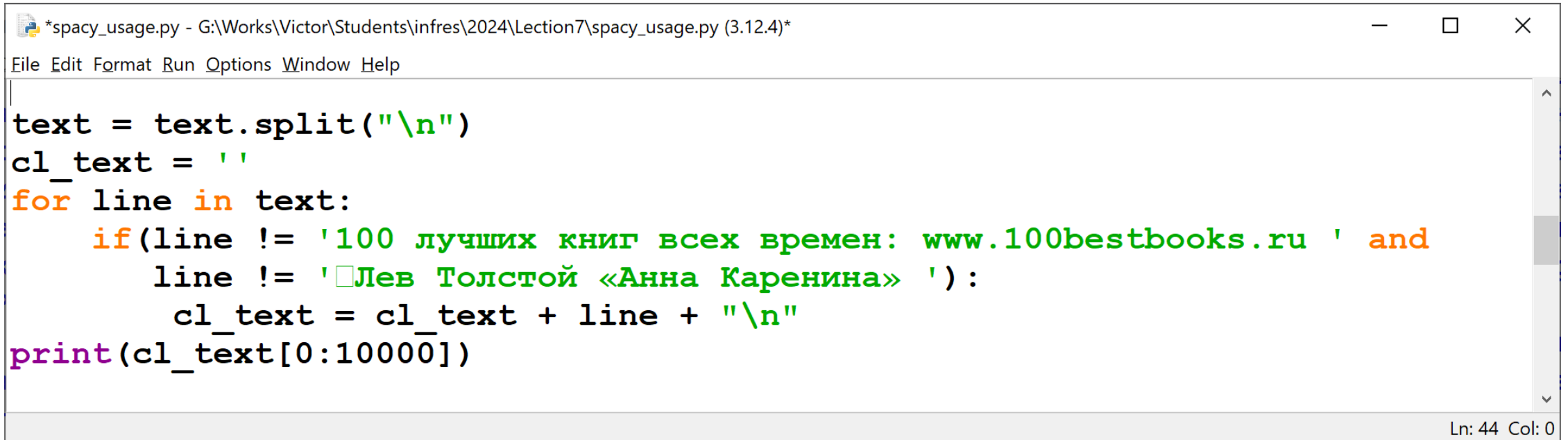
С ним случилось в эту минуту то, что случается с людьми, когда они неожиданно
уличены в чем-нибудь слишком постыдном. Он не сумел приготовить свое лицо к тому
положению, в которое он становился пред женой после открытия его вины. Вместо того чтоб
оскорбиться, отречься, оправдываться, просить прощения, оставаться даже равнодушным –
все было бы лучше того, что он сделал! – его лицо совершенно невольно («рефлексы
головного мозга»2, – подумал Степан Аркадьич, который любил физиологию), совершенно
невольно вдруг улыбнулось привычною, доброю и потому глупою улыбкой.

Эту глупую улыбку он не мог простить себе. Увидав эту улыбку, Долли вздрогнула, как
от физической боли, разразилась, со свойственною ей горячностью, потоком жестоких слов и
```

Ln: 438 Col: 0

# Уберем лишнее 1

---



The screenshot shows a Python IDE window titled '\*spacy\_usage.py - G:\Works\Victor\Students\infres\2024\Lecion7\spacy\_usage.py (3.12.4)\*'. The menu bar includes File, Edit, Format, Run, Options, Window, and Help. The code in the editor is as follows:

```
text = text.split("\n")
cl_text = ''
for line in text:
    if(line != '100 лучших книг всех времен: www.100bestbooks.ru ' and
        line != '☐Лев Толстой «Анна Каренина» '):
        cl_text = cl_text + line + "\n"
print(cl_text[0:10000])
```

The status bar at the bottom right indicates 'Ln: 44 Col: 0'.

# Уберем лишнее 1 (текстом)

---

```
text = text.split("\n")
cl_text = ""
for line in text:
    if(line != '100 лучших книг всех времен: www.100bestbooks.ru ' and
       line != 'Лев Толстой «Анна Каренина» '):
        cl_text = cl_text + line + "\n"
print(cl_text[0:10000])
```

# Шаг 3. Разбор текста

---

sраСу позволяет сделать в одну операцию многие шаги разбора текста, включая токенизацию, тегирование, лемматизацию, нормализацию, удаление стоп слов.

# Шаг 3. Разбор текста

```
spacy_usage.py - G:\Works\Victor\Students\infres\2024\Lecture7\spacy_usage.py (3.12.4)
File Edit Format Run Options Window Help
import spacy
from spacy import displacy
# загрузка предобученной модели на русском языке
nlp = spacy.load("ru_core_news_sm")

# обработка текста
# 500 - для примера, при реальном запуске нужно убрать
doc = nlp(cl_text[0:500])
# токенизация текста
for token in doc:
    # для каждого токена выводится его текст, лемма, часть речи, роль в предложении, форма слова,
    # является ли слово буквенным, является ли слово стоп-словом
    print(token.text, token.lemma_, token.pos_, token.dep_,
          token.shape_, token.is_alpha, token.is_stop)
```

Ln: 72 Col: 6

# Шаг 3. Разбор текста (текстом)

---

```
import spacy
from spacy import displacy

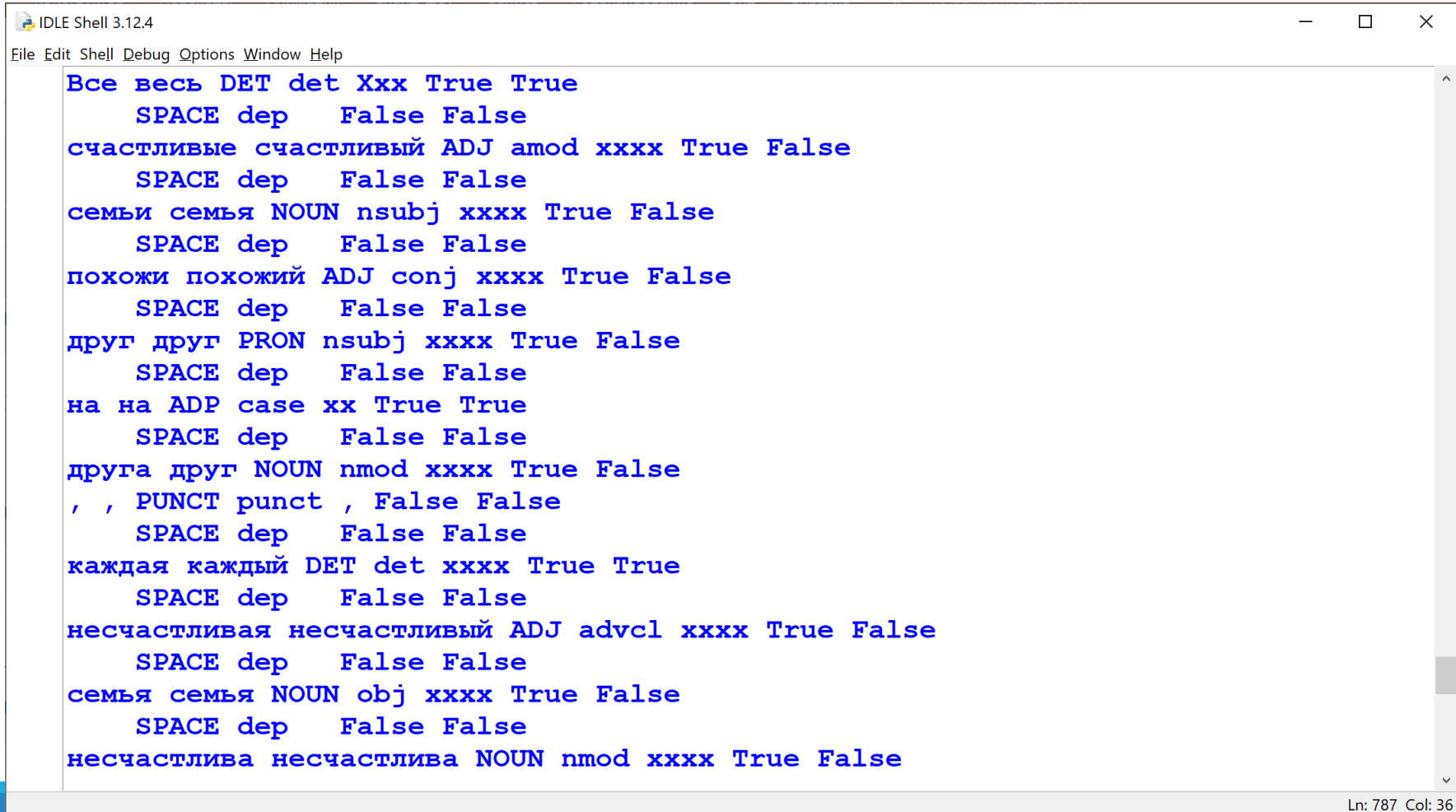
# загрузка предобученной модели на русском языке
nlp = spacy.load("ru_core_news_sm")

# обработка текста
# 500 - для примера, при реальном запуске нужно убрать
doc = nlp(cl_text[0:500])

# токенизация текста
for token in doc:
    # для каждого токена выводится его текст, лемма, часть речи, роль в предложении, форма слова,
    # является ли слово буквенным, является ли слово стоп-словом
    print(token.text, token.lemma_, token.pos_, token.dep_,
          token.shape_, token.is_alpha, token.is_stop)
```



# Шаг 3. Результат (не с начала)



```
IDLE Shell 3.12.4
File Edit Shell Debug Options Window Help
Все весь DET det Xxx True True
SPACE dep False False
счастливые счастливый ADJ amod xxxx True False
SPACE dep False False
семьи семья NOUN nsubj xxxx True False
SPACE dep False False
похожи похожий ADJ conj xxxx True False
SPACE dep False False
друг друг PRON nsubj xxxx True False
SPACE dep False False
на на ADP case xx True True
SPACE dep False False
друга друг NOUN nmod xxxx True False
, , PUNCT punct , False False
SPACE dep False False
каждая каждый DET det xxxx True True
SPACE dep False False
несчастливая несчастливый ADJ advcl xxxx True False
SPACE dep False False
семья семья NOUN obj xxxx True False
SPACE dep False False
несчастлива несчастлива NOUN nmod xxxx True False
```

Ln: 787 Col: 36

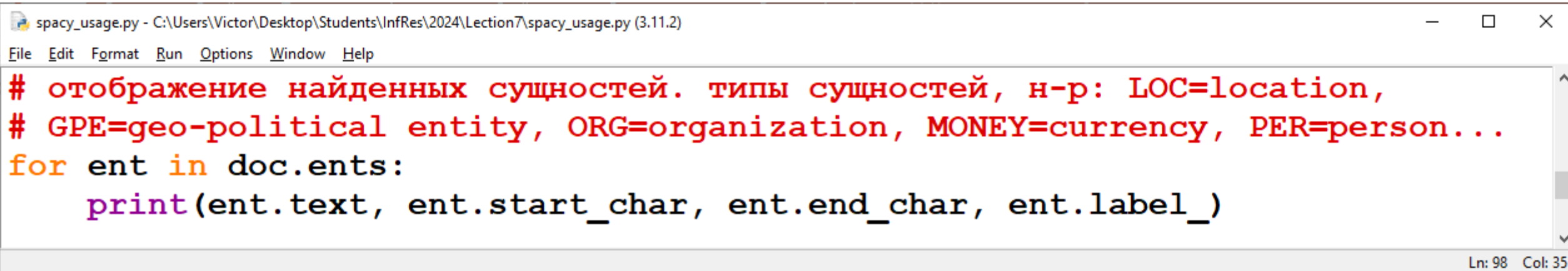
## Шаг 3. Названия и т.д.

---

sраСу умеет классифицировать все нестандартные слова. Например, имена, названия стран, организаций, валют и т.д.

# Шаг 3. Выделение названий

---



The screenshot shows a Python IDE window titled "spacy\_usage.py - C:\Users\Victor\Desktop\Students\InfRes\2024\Lecture7\spacy\_usage.py (3.11.2)". The menu bar includes File, Edit, Format, Run, Options, Window, and Help. The code is written in a monospaced font with syntax highlighting. It consists of two red comment lines, a blue 'for' loop, and a purple 'print' statement. The code iterates over the entities in a document and prints their text, start and end character indices, and their label. The status bar at the bottom right indicates "Ln: 98 Col: 35".

```
spacy_usage.py - C:\Users\Victor\Desktop\Students\InfRes\2024\Lecture7\spacy_usage.py (3.11.2)
File Edit Format Run Options Window Help
# отображение найденных сущностей. типы сущностей, н-р: LOC=location,
# GPE=geo-political entity, ORG=organization, MONEY=currency, PER=person...
for ent in doc.ents:
    print(ent.text, ent.start_char, ent.end_char, ent.label_)
```

Ln: 98 Col: 35

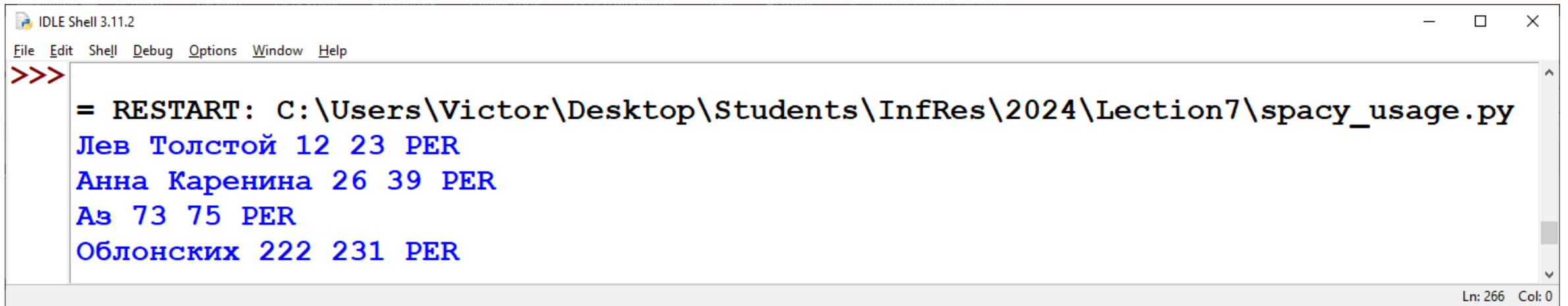
# Шаг 3. Выделение названий

---

```
# отображение найденных сущностей. типы сущностей, н-р: LOC=location,  
# GPE=geo-political entity, ORG=organization, MONEY=currency, PER=person...  
for ent in doc.ents:  
    print(ent.text, ent.start_char, ent.end_char, ent.label_)
```

# Шаг 3. Выделение названий

---



```
IDLE Shell 3.11.2
File Edit Shell Debug Options Window Help
>>> = RESTART: C:\Users\Victor\Desktop\Students\InfRes\2024\Lecture7\spacy_usage.py
Лев Толстой 12 23 PER
Анна Каренина 26 39 PER
Аз 73 75 PER
Облонских 222 231 PER
Ln: 266 Col: 0
```

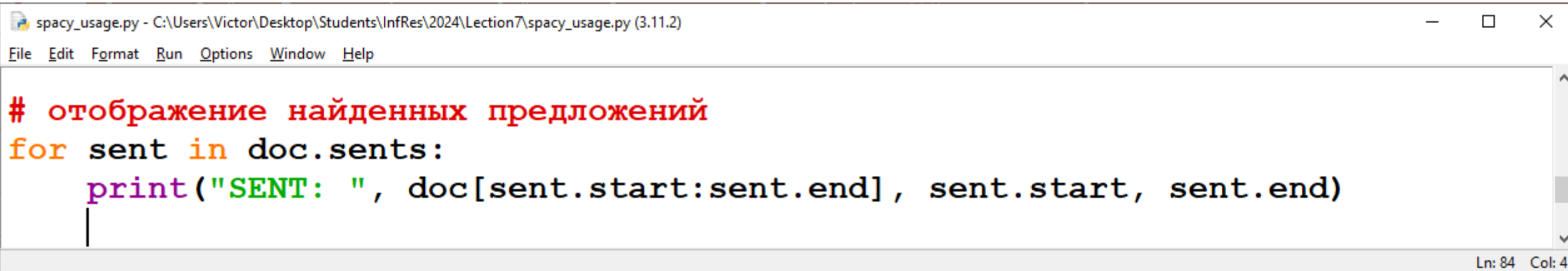
# Шаг 3. Предложения

---

sраСу умеет выделять из текста предложения.

# Шаг 3. Выделение предложений

---



The screenshot shows a Python IDE window titled "spacy\_usage.py - C:\Users\Victor\Desktop\Students\InfRes\2024\Lecture7\spacy\_usage.py (3.11.2)". The menu bar includes File, Edit, Format, Run, Options, Window, and Help. The code in the editor is as follows:

```
# отображение найденных предложений
for sent in doc.sents:
    print("SENT: ", doc[sent.start:sent.end], sent.start, sent.end)
    |
```

The status bar at the bottom right indicates "Ln: 84 Col: 4".

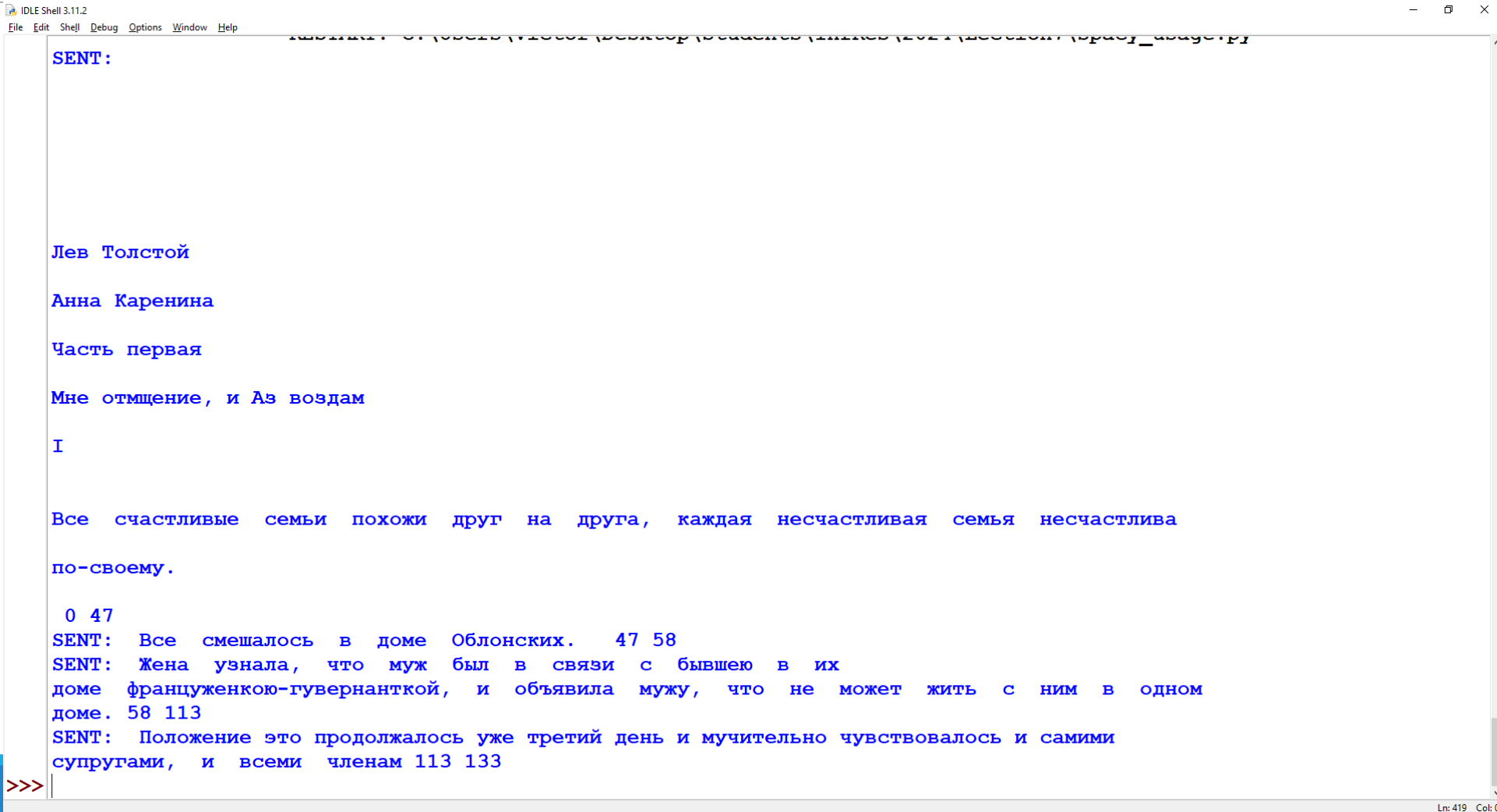
# Шаг 3. Выделение предложений

---

```
# отображение найденных предложений
for sent in doc.sents:
    print("SENT: ", doc[sent.start:sent.end], sent.start, sent.end)
```



# Шаг 3. Выделение предложений



```
IDLE Shell 3.11.2
File Edit Shell Debug Options Window Help
SENT:
Лев Толстой
Анна Каренина
Часть первая
Мне отмщение, и Аз воздам
I
Все счастливые семьи похожи друг на друга, каждая несчастливая семья несчастлива
по-своему.
0 47
SENT: Все смешалось в доме Облонских. 47 58
SENT: Жена узнала, что муж был в связи с бывшею в их
доме француженкою-гувернанткой, и объявила мужу, что не может жить с ним в одном
доме. 58 113
SENT: Положение это продолжалось уже третий день и мучительно чувствовалось и самими
супругами, и всеми членам 113 133
>>>
```

## Шаг 3. Уберем лишнее 2

```
lection7.py - C:\Users\Victor\Desktop\Students\InfRes\2024\Lecture7\lection7.py (3.11.2)
File Edit Format Run Options Window Help

# токенизация текста
tokens = []
for token in doc:
    # для каждого токена выводится его текст, лемма, часть речи, роль в предложе
    # является ли слово буквенным, является ли слово стоп-словом
    if(token.is_alpha and not token.is_stop and token.ent_type == 0 and
        not token.pos_ == 'ADJ'):
        tokens.append(token)
        print(token.text, token.lemma_, token.pos_, token.dep_,
              token.shape_, token.is_alpha, token.is_stop, token.ent_type_)
```

Ln: 82 Col: 77

## Шаг 3. Уберем лишнее 2

---

```
# токенизация текста
```

```
tokens = []
```

```
for token in doc:
```

```
    # для каждого токена выводится его текст, лемма, часть речи, роль в предложении, форма слова,
```

```
    # является ли слово буквенным, является ли слово стоп-словом
```

```
    if(token.is_alpha and not token.is_stop and token.ent_type == 0 and
```

```
       not token.pos_ == 'ADJ'):
```

```
        tokens.append(token)
```

```
        print(token.text, token.lemma_, token.pos_, token.dep_,
```

```
              token.shape_, token.is_alpha, token.is_stop, token.ent_type_)
```

# Результат

```
== RESTART: C:\Users\Victor\Desktop\Students\InfRes\2024\Lecture7\lecture7.py ==
Часть часть NOUN appos Xxxxx True False
отмщение отмщение NOUN ROOT xxxx True False
воздам воздам NOUN nmod xxxx True False
семьи семья NOUN nsubj xxxx True False
друг друг PRON obl xxxx True False
друга друга PRON fixed xxxx True False
семья семья NOUN conj xxxx True False
несчастлива несчастлива NOUN nmod xxxx True False
смешалось смешаться VERB ROOT xxxx True False
доме дом NOUN obl xxxx True False
Жена жена NOUN nsubj Xxxx True False
узнала узнать VERB ROOT xxxx True False
муж муж NOUN nsubj xxx True False
связи связь NOUN fixed xxxx True False
бывшего бывшею NOUN obl xxxx True False
доме дом NOUN nmod xxxx True False
француженкою француженкою NOUN appos xxxx True False
гувернанткой гувернантка NOUN appos xxxx True False
объявила объявить VERB ccomp xxxx True False
мужу муж NOUN iobj xxxx True False
жить жить VERB xcomp xxxx True False
доме дом NOUN obl xxxx True False
Положение положение NOUN nsubj Xxxxx True False
продолжалось продолжаться VERB ROOT xxxx True False
день день NOUN obl xxxx True False
мучительно мучительно ADV advmod xxxx True False
чувствовалось чувствоваться VERB conj xxxx True False
супругами супругами NOUN obl xxxx True False
планам план NOUN conj xxxx True False
```

# Шаг 4. Вычисление количества вхождений СЛОВ

```
lecture7.py - C:\Users\Victor\Desktop\Students\InfRes\2024\Lecture7\lecture7.py (3.11.2)
File Edit Format Run Options Window Help
# Расчёт частот слов
words = {}
i = 0
words_counter = []
words_counter_x = []
words_counter_y = []
for token in tokens:
    if token.lemma_ in words:
        words[token.lemma_] += 1
    else:
        words[token.lemma_] = 1
    if not token.lemma_ in words_counter:
        words_counter.append(token.lemma_)
        words_counter_x.append(len(words_counter))
        words_counter_y.append(i)
    i += 1
points_y = sorted(list(words.values()), reverse = True)
points_x = list(range(0, len(points_y)))
```

Ln: 98 Col: 37

# Шаг 4. Вычисление количества вхождений СЛОВ

---

```
# Расчёт частот слов
```

```
words = {}
```

```
i = 0
```

```
words_counter = []
```

```
words_counter_x = []
```

```
words_counter_y = []
```

```
for token in tokens:
```

```
    if token.lemma_ in words:
```

```
        words[token.lemma_] += 1
```

```
    else:
```

```
        words[token.lemma_] = 1
```

```
if not token.lemma_ in words_counter:
```

```
    words_counter.append(token.lemma_)
```

```
    words_counter_x.append(len(words_counter))
```

```
    words_counter_y.append(i)
```

```
    i += 1
```

```
points_y = sorted(list(words.values()), reverse = True)
```

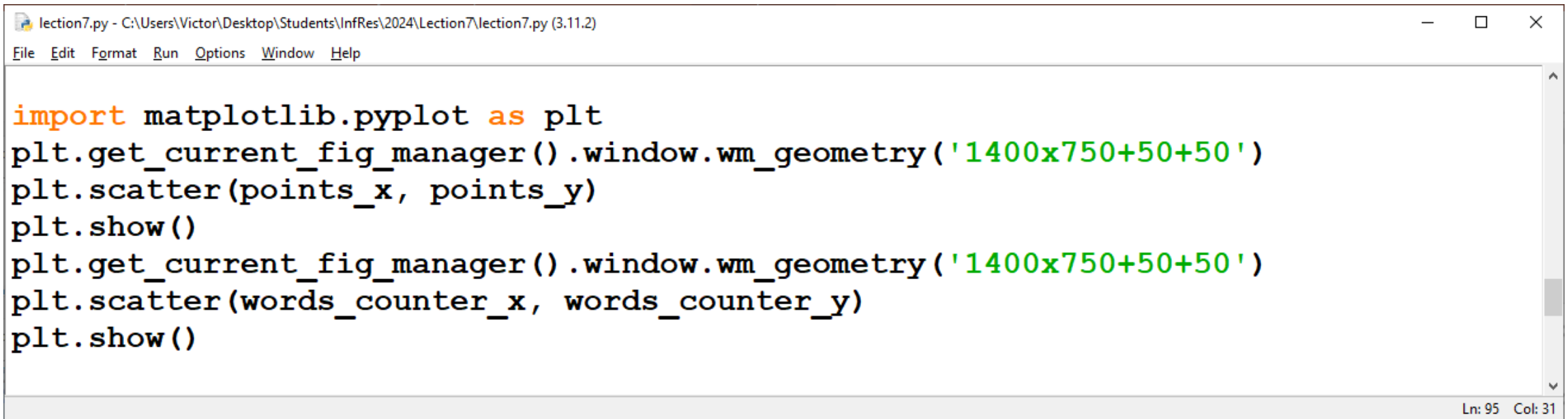
```
points_x = list(range(0, len(points_y)))
```

# Результат

```
IDLE Shell 3.11.2
File Edit Shell Debug Options Window Help
>>>
== RESTART: C:\Users\Victor\Desktop\Students\InfRes\2024\Lecture7\lecture7.py ==
{'часть': 63, 'отмщение': 1, 'воздам': 1, 'семья': 63, 'друг': 149, 'друга': 37,
 'несчастлива': 2, 'смешаться': 6, 'дом': 232, 'жена': 385, 'узнать': 166, 'муж':
 : 336, 'связь': 60, 'бывшею': 1, 'француженкою': 1, 'гувернантка': 23, 'объявить
 ': 37, 'жить': 242, 'положение': 306, 'продолжаться': 29, 'день': 298, 'мучитель
 но': 27, 'чувствоваться': 12, 'супругами': 2, 'член': 50, 'домочадец': 4, 'чувст
 вовать': 479, 'смысл': 56, 'сожителство': 1, 'каждый': 67, 'двор': 46, 'случайн
 о': 8, 'сошедшиеся': 3, 'человек': 694, 'связать': 30, 'выходить': 97, 'комната'
 : 176, 'дома': 34, 'ребёнок': 363, 'бегать': 9, 'англичанка': 26, 'поссориться':
 5, 'экономка': 4, 'написать': 73, 'записка': 66, 'приятельница': 13, 'прося': 9
 , 'приискать': 1, 'место': 190, 'повар': 11, 'уйти': 84, 'вчера': 91, 'время': 4
 50, 'обед': 117, 'кухарка': 3, 'кучер': 37, 'просить': 130, 'расчёт': 21, 'ссора
 ': 25, 'князь': 157, 'звать': 50, 'свет': 163, 'час': 167, 'восемь': 23, 'утро':
 94, 'проснуться': 30, 'спальня': 34, 'кабинет': 84, 'диван': 28, 'вернуть': 1
 3, 'тело': 55, 'пружина': 7, 'желать': 226, 'заснуть': 41, 'надолго': 9, 'сторон
 а': 187, 'крепко': 24, 'обнять': 23, 'подушка': 16, 'прижаться': 4, 'щека': 25,
 'вскочить': 25, 'сесть': 112, 'открыть': 56, 'глаз': 471, 'думать': 648, 'вспоми
 нать': 101, 'сон': 42, 'давать': 110, 'стол': 169, 'петь': 18, 'il': 8, 'mio': 2
 , 'tesoro': 1, 'графинчик': 3, 'женщина': 255, 'весело': 66, 'заблестели': 5, 'з
 адуматься': 29, 'улыбаться': 217, 'скажешь': 11, 'слово': 396, 'мысль': 335, 'на
 яву': 3, 'выразить': 33, 'заметить': 141, 'полоса': 7, 'пробиться': 1, 'сбоку':
 4, 'стор': 1, 'скинуть': 3, 'нога': 152, 'отыскать': 4, 'шитые': 1, 'подарок': 9
 , 'рождение': 10, 'год': 194, 'обделать': 1, 'золотистый': 1, 'сафьян': 1, 'туфл
 я': 7, 'привычка': 47, 'потянуться': 5, 'рука': 715, 'висеть': 6, 'халат': 9, 'в
 спомнить': 148, 'спать': 90, 'улыбка': 263, 'исчезнуть': 26, 'лицо': 611, 'сморщ
 ить': 4, 'лоб': 33, 'aaa': 4, 'замычать': 3, 'воображение': 31, 'представиться':
 16, 'подробность': 83, 'женою': 28, 'безвыходность': 3, 'вина': 35, 'простить':
 82, 'драма': 9, 'приговаривал': 3, 'отчаяние': 47, 'впечатление': 82, 'неприятн
```

# Шаг 4. Построение графиков

---

A screenshot of a Python IDE window titled 'lection7.py - C:\Users\Victor\Desktop\Students\InfRes\2024\Lection7\lection7.py (3.11.2)'. The window has a menu bar with 'File', 'Edit', 'Format', 'Run', 'Options', 'Window', and 'Help'. The main text area contains the following Python code:

```
import matplotlib.pyplot as plt
plt.get_current_fig_manager().window.wm_geometry('1400x750+50+50')
plt.scatter(points_x, points_y)
plt.show()
plt.get_current_fig_manager().window.wm_geometry('1400x750+50+50')
plt.scatter(words_counter_x, words_counter_y)
plt.show()
```

The code is color-coded: 'import' is orange, 'matplotlib.pyplot' is blue, 'as plt' is orange, and the window geometry strings are green. The status bar at the bottom right shows 'Ln: 95 Col: 31'.

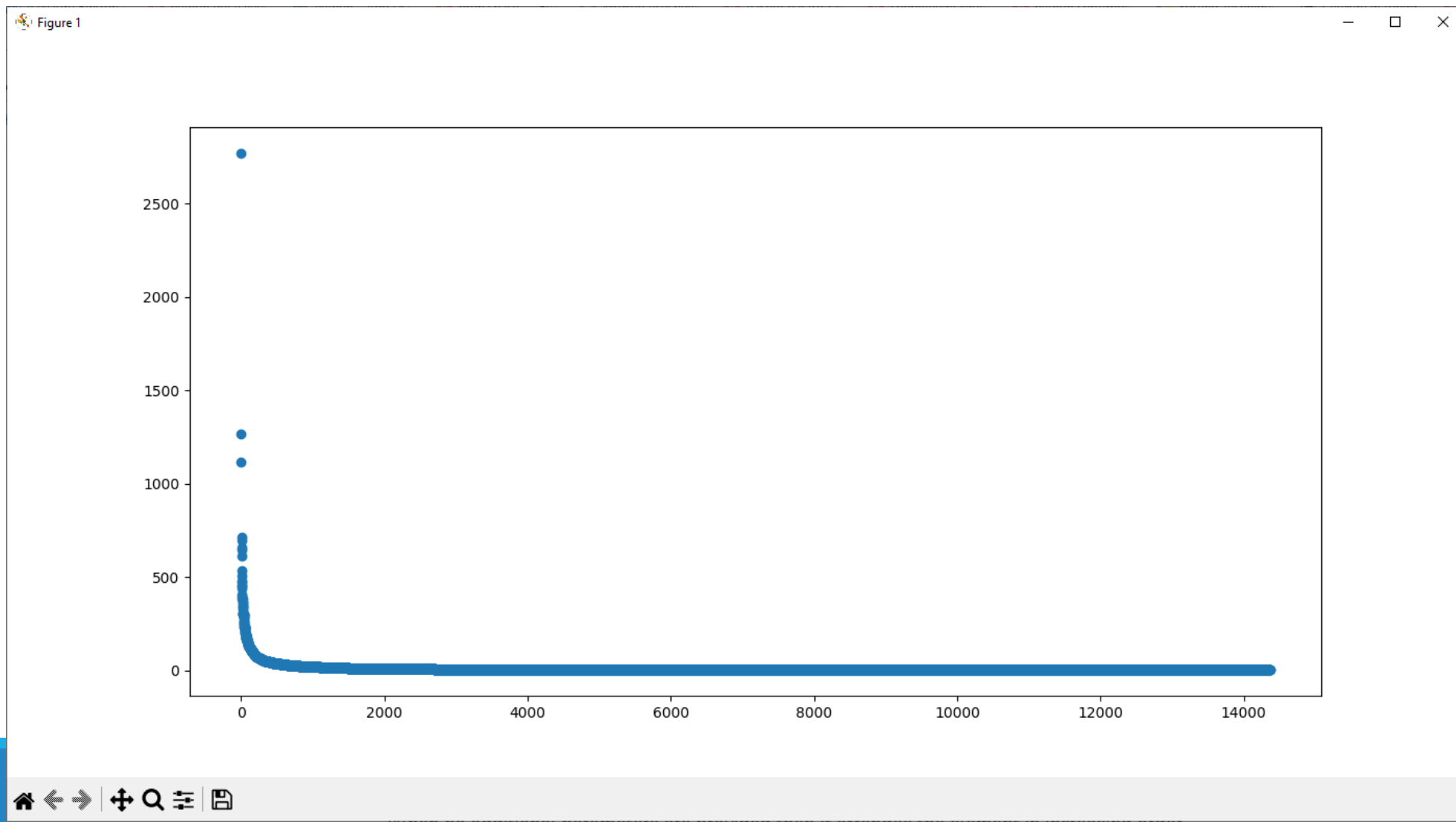


# Шаг 4. Построение графиков

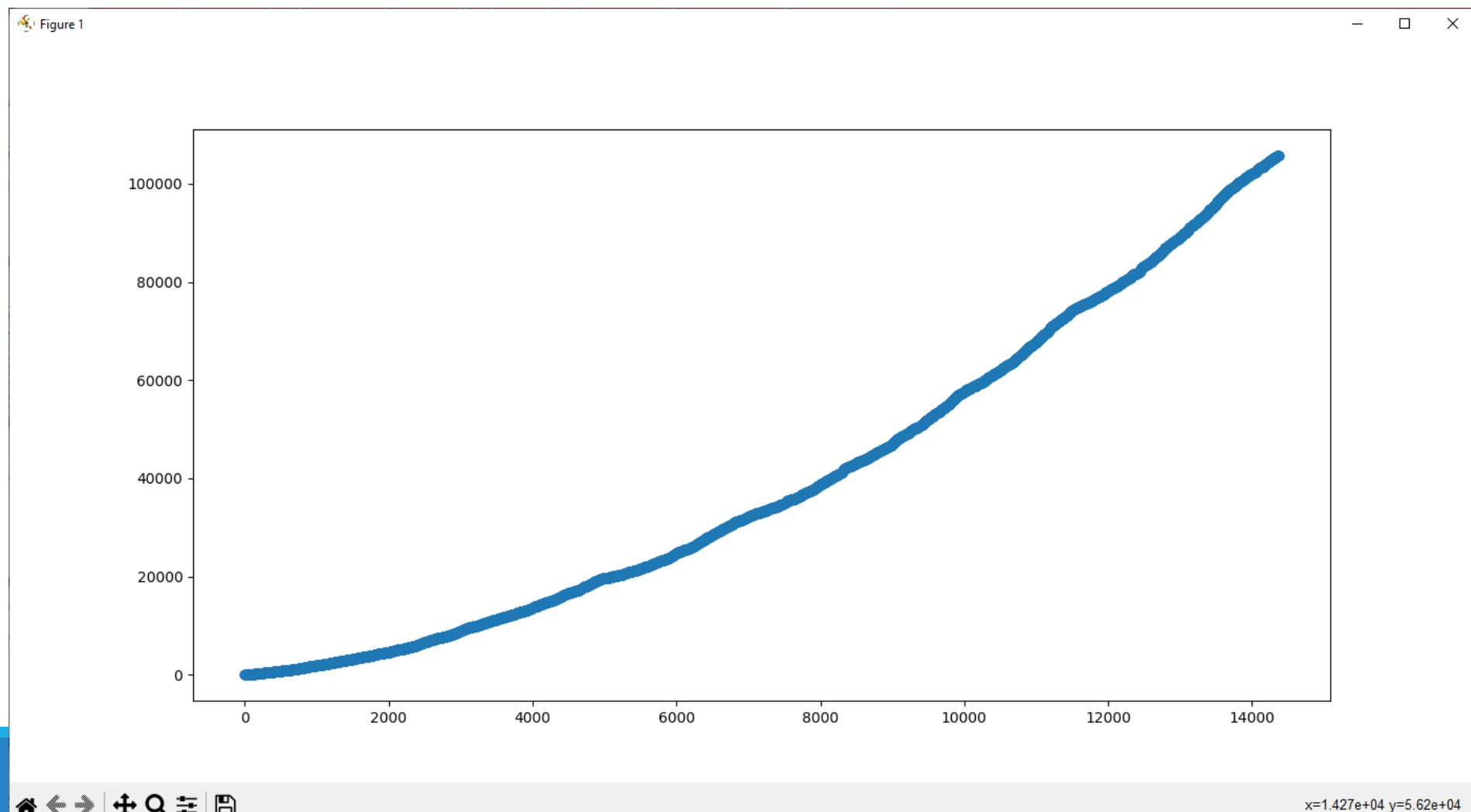
---

```
import matplotlib.pyplot as plt  
  
plt.get_current_fig_manager().window.wm_geometry('1400x750+50+50')  
plt.scatter(points_x, points_y)  
plt.show()  
  
plt.get_current_fig_manager().window.wm_geometry('1400x750+50+50')  
plt.scatter(words_counter_x, words_counter_y)  
plt.show()
```

# Результат



# Результат



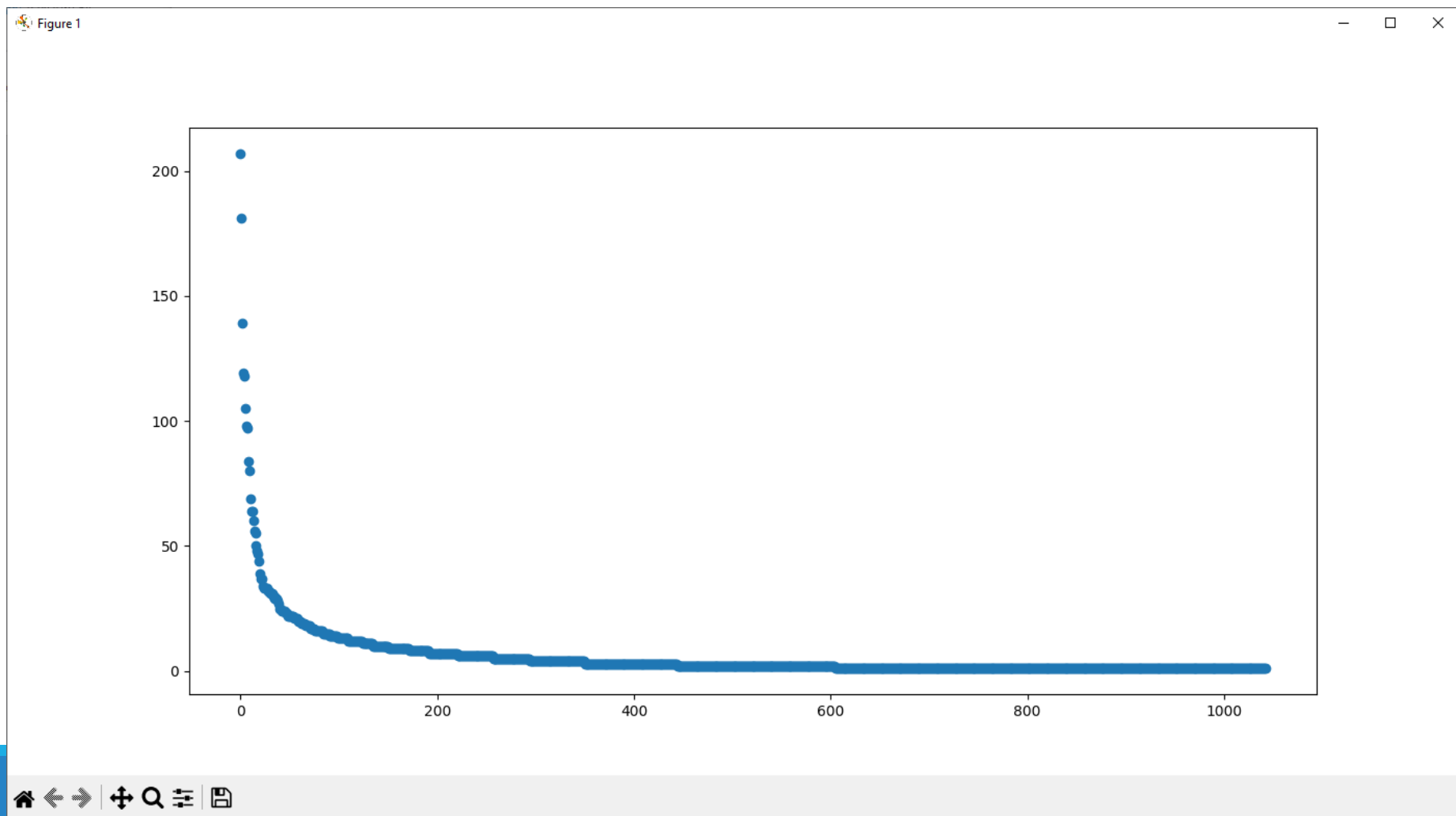
# Другая задача для примеров

---

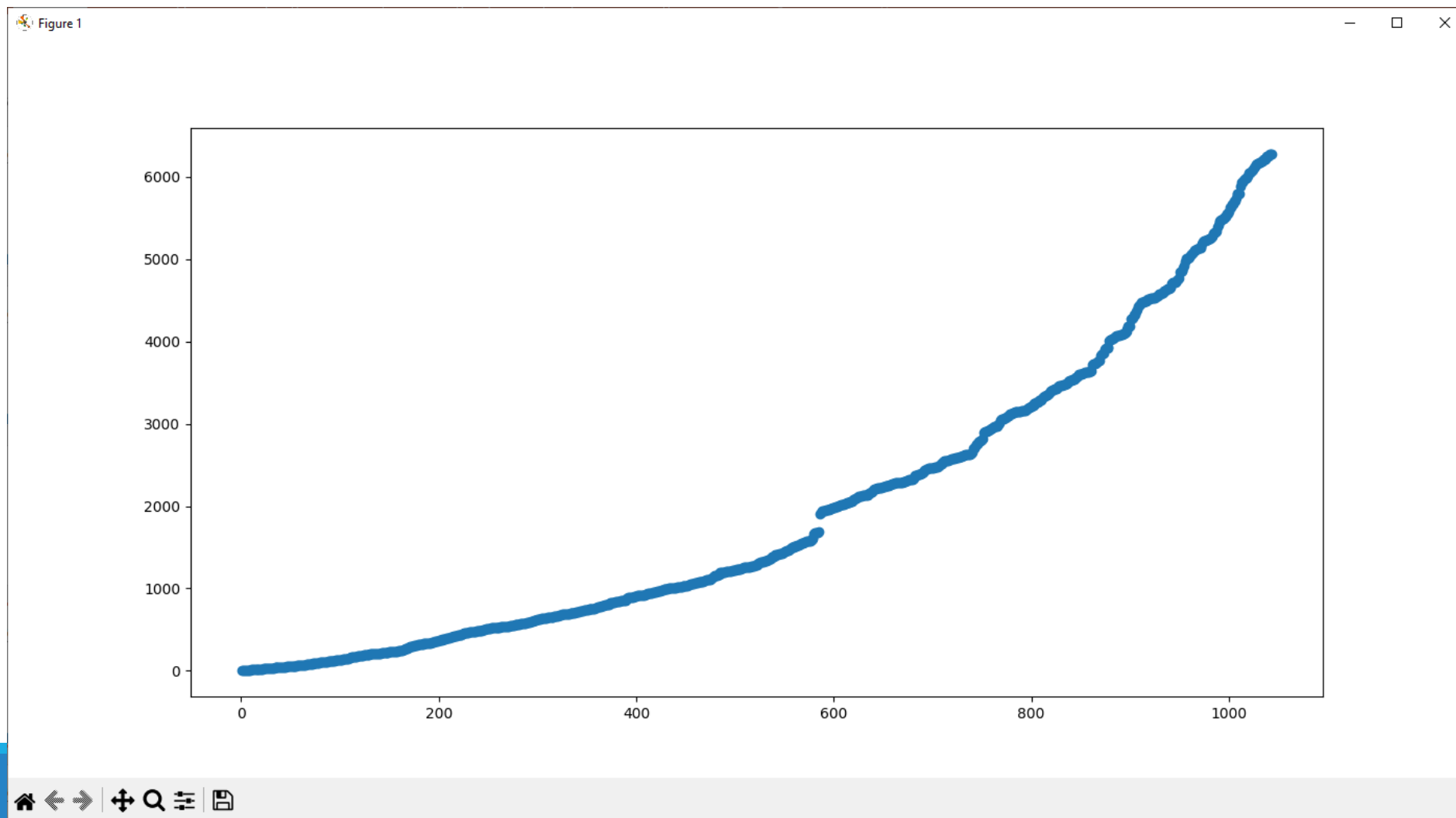
Построим дискретную функцию зависимости числа неповторяющихся слов (словаря) от общего числа слов текста для Конституции Российской Федерации:

<https://constitutionrf.ru/constitutionrf.pdf?ysclid=m8yd89g8dl404010678>

# Результат



# Результат



# Определение семантической близости

---

Библиотека `sraCy` также предоставляет механизмы для сравнения текстов на семантическую близость. Для этого есть метод `similarity` (используется механизм, аналогичный `Word2Vec`).

# Определение семантической близости

```
lection7sem.py - C:\Users\Victor\Desktop\Students\InfRes\2024\Lecture7\lection7sem.py (3.11.2)
File Edit Format Run Options Window Help

import spacy
from spacy import displacy
# загрузка предобученной модели на русском языке

nlp = spacy.load("ru_core_news_sm")

doc1 = nlp("Как найти деканат?")
doc2 = nlp("Где находится деканат?")
doc3 = nlp("Где A100?")
print(doc1.similarity(doc2))
print(doc1.similarity(doc3))
print(doc2.similarity(doc3))|
```

Ln: 12 Col: 28



# Определение семантической близости

---

```
import spacy
from spacy import displacy
# загрузка предобученной модели на русском языке

nlp = spacy.load("ru_core_news_sm")

doc1 = nlp("Как найти деканат?")
doc2 = nlp("Где находится деканат?")
doc3 = nlp("Где A100?")
print(doc1.similarity(doc2))
print(doc1.similarity(doc3))
print(doc2.similarity(doc3))
```

# Результат

```
IDLE Shell 3.11.2
File Edit Shell Debug Options Window Help

sult of the Doc.similarity method will be based on the tagger, parser and NER, w
hich may not give useful similarity judgements. This may happen if you're using
one of the small models, e.g. `en_core_web_sm`, which don't ship with word vecto
rs and only use context-sensitive tensors. You can always add your own word vect
ors, or use one of the larger models instead if available.
0.4869630067605179

Warning (from warnings module):
  File "C:\Users\Victor\Desktop\Students\InfRes\2024\Lecture7\lecture7sem.py", l
ine 11
    print(doc1.similarity(doc3))
UserWarning: [W007] The model you're using has no word vectors loaded, so the re
sult of the Doc.similarity method will be based on the tagger, parser and NER, w
hich may not give useful similarity judgements. This may happen if you're using
one of the small models, e.g. `en_core_web_sm`, which don't ship with word vecto
rs and only use context-sensitive tensors. You can always add your own word vect
ors, or use one of the larger models instead if available.
0.2774371075270976

Warning (from warnings module):
  File "C:\Users\Victor\Desktop\Students\InfRes\2024\Lecture7\lecture7sem.py", l
ine 12
    print(doc2.similarity(doc3))
UserWarning: [W007] The model you're using has no word vectors loaded, so the re
sult of the Doc.similarity method will be based on the tagger, parser and NER, w
hich may not give useful similarity judgements. This may happen if you're using
one of the small models, e.g. `en_core_web_sm`, which don't ship with word vecto
rs and only use context-sensitive tensors. You can always add your own word vect
ors, or use one of the larger models instead if available.
0.5839445907967953
>>>
```

# Полезные ссылки

---

1. <https://spacy.io/>
2. [https://spacy.io/models/ru#ru\\_core\\_news\\_sm](https://spacy.io/models/ru#ru_core_news_sm)