



Міністерство освіти і науки України

Національний технічний університет України

“Київський політехнічний інститут імені Ігоря Сікорського”

Факультет інформатики та обчислювальної техніки

Кафедра інформатики та програмної інженерії

Лабораторна робота №8

Аналіз даних

Тема: Аналіз текстів.

Виконав

студент групи ІІІ-11:

Панченко С. В.

Перевірила:

Олійник Ю. О

Київ 2023

ЗМІСТ

1 Мета лабораторної роботи.....	6
2 Завдання.....	7
3 Виконання.....	8
3.1 Основне завдання.....	8
3.2 Додаткове завдання 1.....	16
3.3 Додаткове завдання 2.....	23
ДОДАТОК А ТЕКСТИ ПРОГРАМНОГО КОДУ.....	27

1 МЕТА ЛАБОРАТОРНОЇ РОБОТИ

Ознайомитись з методом аналізу текстів.

2 ЗАВДАННЯ

Дані для виконання: текстові дані у форматі csv-файлів або дані з відкритих джерел

(телеграм-канали, RSS-канали тощо). Приклад даних за [посиланням](#)

1. Нормалізація та попередня обробка даних.

1. провести очищення текстових даних від стоп-слів/тегів/розмітки;
2. виконати токенізацію текстових елементів;
3. провести лематизацію текстових елементів (можна використати бібліотеку Spacy - приклад роботи за [посиланням](#)). Зберегти результат в окремий файл.

4. Створити Bag of Words для всіх нормалізованих слів. Зберегти результат в окремий файл.

5. Порахувати метрику TF-IDF для 10 слів, що найчастіше зустрічаються в корпусі;

Додаткове завдання

1. Інтелектуальний аналіз текстів (+1 бал):

- провести сантисмент аналіз (визначення емоційної тональності – позитивний / негативний) для даних [ukr_text.csv](#). Для визначення тональності можна використати як методи на основі [словника тональності](#) (посилання) так і методи машинного навчання.

- провести категоризацію (визначення категорій тексту) даних методом LSA. Приклади роботи з морфоаналізатором PyMorphu наведено за [посиланням](#).

2. Обробка даних оповідань А.К. Дойля та Е.По (+1 бал):

- [Завантажити потрібні дані](#).
- Завантажити оповідання А.К. Дойля та Е.По з папки Texts/Task.
- Виконати попередню обробку текстів.
- Побудувати дві хмари слів, що використовують А.К. Дойль та Е.По.
- Який з письменників написав більш похмурі оповідання?

3 ВИКОНАННЯ

3.1 Основне завдання

```
In [1]: import numpy as np
import pandas as pd
import nltk
import re
df = pd.read_csv('bbc-news-data.csv', sep='\t')
df
```

Out[1]:

	category	filename	title	content
0	business	001.txt	Ad sales boost Time Warner profit	Quarterly profits at US media giant TimeWame...
1	business	002.txt	Dollar gains on Greenspan speech	The dollar has hit its highest level against ...
2	business	003.txt	Yukos unit buyer faces loan claim	The owners of embattled Russian oil giant Yuk...
3	business	004.txt	High fuel prices hit BA's profits	British Airways has blamed high fuel prices f...
4	business	005.txt	Pernod takeover talk lifts Domecq	Shares in UK drinks and food firm Allied Dome...
...
2220	tech	397.txt	BT program to beat dialler scams	BT is introducing two initiatives to help bea...
2221	tech	398.txt	Spam e-mails tempt net shoppers	Computer users across the world continue to i...
2222	tech	399.txt	Be careful how you code	A new European directive could put software w...
2223	tech	400.txt	US cyber security chief resigns	The man making sure US computer networks are ...
2224	tech	401.txt	Losing yourself in online gaming	Online role playing games are time-consuming,...

2225 rows × 4 columns

Рисунок 3.1.1 - Зчитування файлу

Видалимо колонки 'filename', 'title', 'category'.

```
In [2]: df.drop(['filename', 'title', 'category'], axis=1, inplace=True)
df
```

Out[2]:

	content
0	Quarterly profits at US media giant TimeWame...
1	The dollar has hit its highest level against ...
2	The owners of embattled Russian oil giant Yuk...
3	British Airways has blamed high fuel prices f...
4	Shares in UK drinks and food firm Allied Dome...
...	...
2220	BT is introducing two initiatives to help bea...
2221	Computer users across the world continue to i...
2222	A new European directive could put software w...
2223	The man making sure US computer networks are ...
2224	Online role playing games are time-consuming,...

2225 rows × 1 columns

Рисунок 3.1.2 - Видалення колонок

Видалимо порожні документи, якщо вони є.

```
In [3]: df = df[~(df.content.str.strip() == '')]
df
```

```
Out[3]:
```

	content
0	Quarterly profits at US media giant TimeWarne...
1	The dollar has hit its highest level against ...
2	The owners of embattled Russian oil giant Yuk...
3	British Airways has blamed high fuel prices f...
4	Shares in UK drinks and food firm Allied Dome...

Рисунок 3.1.3 - Видалення порожніх документів

Визначимо стоп-слова англійської мови.

```
In [4]: wpt = nltk.WordPunctTokenizer()
stop_words = nltk.corpus.stopwords.words('english')
```

Рисунок 3.1.4 - Стоп-слова

Визначимо функцію, що виконує попередню обробку документу. Застосуємо декоратор `pr.vectorize` для того, щоб функція могла працювати з корпусами.

```
In [5]: @np.vectorize
def preproc_doc(doc):
    doc = re.sub(r'^a-zA-Z\s|', '', doc, re.I | re.A)
    doc = doc.lower()
    doc = doc.strip()
    tokens = wpt.tokenize(doc)
    filtered_tokens = [token for token in tokens if token not in stop_words]
    doc = ' '.join(filtered_tokens)
    return doc

p_corpus = preproc_doc(df.content)
p_corpus
```

Out[5]: array(['quarterly profits us media giant timewarner jumped bn three months december yearea
rlier firm one biggest investors google benefited sales highspeed internet connections hig
her advert sales timewarner said fourth quarter sales rose bn bn profits buoyed oneoff gai
ns offset profit dip warner bros less users aol time warner said friday owns searchengine
google internet business aol mixed fortunes lost subscribers fourth quarter profits lower
preceding three quarters however company said aols underlying profit exceptional items ros
e back stronger internet advertising revenues hopes increase subscribers offering online s
ervice free timewarner internet customers try sign aols existing customers highspeed broad
band timewarner also restate results following probe us securities exchange commission sec
close concluding time warners fourth quarter profits slightly better analysts expectations
film division saw profits slump helped boxoffice flops alexander catwoman sharp contrast y
earearlier third final film lord rings trilogy boosted results fullyear timewarner posted
profit bn performance revenues grew bn financial performance strong meeting exceeding full
year objectives greatly enhancing flexibility chairman chief executive richard parsons sai
d timewarner projecting operating earnings growth around also expects higher revenue wider
profit margins timewarner restate accounts part efforts resolve inquiry aol us market regu
lators already offered pay settle charges deal review sec company said unable estimate amo
unt needed set aside legal reserves previously set intends adjust way accounts deal german
music publisher bertelsmanns purchase stake aol europe reported advertising revenue book s
ale stake aol europe loss value stake',
'dollar hit highest level euro almost three months federal reserve head said us tra
de deficit set stabilise alan greenspan highlighted us governments willingness curb spendi
ng rising household savings factors may help reduce late trading new york dollar reached e
uro thursday market concerns deficit hit greenback recent months friday federal reserve ch
airman mr greenspans speech london ahead meeting g finance ministers sent dollar higher ea
rlier tumbled back worse than expected us jobs data think chairmans taking much sanguine vie
w current account deficit hes taken time said robert sinche head currency strategy bank am
erica new york hes taking longer term view laying set conditions current account deficit im
prove year next worries deficit concerns china however remain chinas currency remains pegg
ed dollar us currencys sharp falls recent months therefore made chinese export prices high
ly competitive calls shift Beijings policy fallen deaf ears despite recent comments major
chinese newspaper time ripe loosening peg g meeting thought unlikely produce meaningful mo
vement chinese policy meantime us federal reserves decision february boost interest rates
quarter point sixth move many months opened differential european rates halfpoint window b
elieve could enough keep us assets looking attractive could help prop dollar recent falls
partly result big budget deficits well us yawning current account gap need funded buying
us bonds assets foreign firms governments white house announce budget monday many commenta
tors believe deficit remain close half trillion dollars',
'owners embattled russian oil giant yukos ask buyer former production unit pay back
loan stateowned rosneft bought yugansk unit bn sale forced russia part settle bn tax claim
yukos yukos owner menatep group says ask rosneft repay loan yugansk secured assets rosneft
already faces similar repayment demand foreign banks legal experts said rosnefts purchase
yugansk would include obligations pledged assets rosneft pay real money creditors avoid se
izure yugansk assets said moscowbased us lawyer jamie firestone connected case menatep gro
ups managing director tim osborne told reuters news agency default fight rule law exists i
nternational arbitration clauses credit rosneft officials unavailable comment company said
intends take action menatep recover tax claims debts owed yugansk yukos filed bankruptcy p
rotection us court attempt prevent forced sale main production arm sale went ahead decembe
r yugansk sold littleknown shell company turn bought rosneft yukos claims downfall punishm
ent political ambitions founder mikhail khodorkovsky vowed sue participant sale',
...
'new european directive could put software writers risk legal action warns former p
rogrammer technology analyst bill thompson gets way dutch government conclude presidency e
uropean union pushing controversial measure rejected european parliament lacks majority su
pport national governments leave millions european citizens legal limbo facing possibility
court cases new law border controls defence even new constitution tv screens would full ex
perts agonising impact daily lives sadly directly affected controversy concerns patenting
computer programs topic may excite bloggers campaigning groups technical press obsess midd
le britain much fuss generate directive patentability computerimplemented inventions way a
mend article european patent convention yet new directive nodded next meeting one eus min
isterial councils seems likely allow programs patented europe us many observers computing
scene including think results disastrous small companies innovative programmers free open
source software movement let large companies patent sorts ideas give legal force want limi
t competitors use really obvious ideas us cannot build system stores customer credit card
details pay without reenter unless amazon lets hold patent oneclick online purchase small
invention amazon made patent office first owns relatively free sort thing perhaps long new
proposals go back although argument patentability software computerimplemented inventions
going since least mids come head year proposals made endorsed council ministers radically
modified european parliament represented original form national governments seem aware pro
blems poland rejected proposal germanys main political parties opposed enough opposition g
uarantee rejection early december british government held consultation meeting commented p

Рисунок 3.1.5 - Обробка документів

Розіб'ємо кожний документ на окремі слова, об'єднаємо усі слова в одну сукупність.

```
In [6]: words = []
        for doc in p_corpus:
            words.extend(doc.split(' '))
        df_words = pd.DataFrame(set(words))
        df_words.columns = ['words']
        df_words.head(10)
```

```
Out[6]:
```

	words
0	pompeys
1	salaam
2	comfortable
3	turin
4	manoeuvring
5	truck
6	mayfield
7	educate
8	incensed
9	howards

Рисунок 3.1.6 - Розбиття на слова

Визначимо частину мови для кожного слова. Використаємо функцію `nltk.pos_tag`.

```
In [7]: df_words['ps'] = [tag for _, tag in nltk.pos_tag(df_words.words)]
        df_words
```

```
Out[7]:
```

	words	ps
0	pompeys	NNS
1	salaam	VBP
2	comfortable	JJ
3	turin	NN
4	manoeuvring	VBG
...
31333	pioneers	NNS
31334	illegals	VBP
31335	fairway	JJ
31336	induction	NN
31337	snowboards	NNS

31338 rows × 2 columns

Рисунок 3.1.7 - Визначення частини мови

Узагальнимо частини мови до звичайних: noun, adjective, verb тощо.


```
In [8]: from nltk.tag import map_tag
df_words['sps'] = [map_tag('en-ptb', 'universal', tag) for tag in df_words.ps]
df_words
```

```
Out[8]:
```

	words	ps	sps
0	pompeys	NNS	NOUN
1	salaam	VBP	VERB
2	comfortable	JJ	ADJ
3	turin	NN	NOUN
4	manoeuvring	VBG	VERB
...
31333	pioneers	NNS	NOUN
31334	illegals	VBP	VERB
31335	fairway	JJ	ADJ
31336	induction	NN	NOUN
31337	snowboards	NNS	NOUN

31338 rows × 3 columns

Рисунок 3.1.8 - Узагальнення чатин мов

Перетворимо узагальнені частини мови на аббревіатури для лематизації.

```
In [9]: abbr = {'NOUN': 'n', 'VERB': 'v', 'ADJ': 'a', 'ADV': 'r'}
def to_abbr(el):
    res = abbr.get(el)
    if res is not None:
        return res
    return ''

df_words['abbr'] = [to_abbr(x) for x in df_words.sps]
df_words
```

```
Out[9]:
```

	words	ps	sps	abbr
0	pompeys	NNS	NOUN	n
1	salaam	VBP	VERB	v
2	comfortable	JJ	ADJ	a
3	turin	NN	NOUN	n
4	manoeuvring	VBG	VERB	v
...
31333	pioneers	NNS	NOUN	n
31334	illegals	VBP	VERB	v
31335	fairway	JJ	ADJ	a
31336	induction	NN	NOUN	n
31337	snowboards	NNS	NOUN	n

31338 rows × 4 columns

Рисунок 3.1.9 - Приведення загальних частин мов до аббревіатур

Проведемо лематизацію кожного слова за допомогою методу `lemmatize` об'єкта класу `nltk.stem.WordNetLemmatizer`.

```
In [10]: from nltk.stem import WordNetLemmatizer
wlem = WordNetLemmatizer()
def my_lem(word, abbr):
    if abbr == '':
        return wlem.lemmatize(word)
    return wlem.lemmatize(word, pos=abbr)
df_words['lemms'] = [my_lem(row[1]['words'], row[1]['abbr'])
                     for row in df_words.loc[:, ['words', 'abbr']].iterrows()]
df_words
```

```
Out[10]:
```

	words	ps	sps	abbr	lemms
0	pompeys	NNS	NOUN	n	pompey
1	salaam	VBP	VERB	v	salaam
2	comfortable	JJ	ADJ	a	comfortable
3	turin	NN	NOUN	n	turin
4	manoeuvring	VBG	VERB	v	manoeuvre
...
31333	pioneers	NNS	NOUN	n	pioneer
31334	illegals	VBP	VERB	v	illegals
31335	fairway	JJ	ADJ	a	fairway
31336	induction	NN	NOUN	n	induction
31337	snowboards	NNS	NOUN	n	snowboard

31338 rows × 5 columns

Рисунок 3.1.10 - Лематизація слів

Представимо корпус як модуль "Сумка слів". Використаємо для цього клас CountVectorizer зі sklearn.feature_extraction.text.

```
In [11]: from sklearn.feature_extraction.text import CountVectorizer
cv = CountVectorizer(min_df=0., max_df=1.)
cv_matrix = cv.fit_transform(p_corpus)
cv_matrix = pd.DataFrame(cv_matrix.toarray(),
                        columns=cv.get_feature_names_out())
cv_matrix.to_csv('bag_of_words.csv')
```

Рисунок 3.1.11 - Сумка слів

Представимо корпус як модель TD-IDF. Перетворимо матрицю з частотою термінів на матрицю tfidf.

```
In [12]: from sklearn.feature_extraction.text import TfidfTransformer
         tt = TfidfTransformer(norm='l2', use_idf=True)
         tt_matrix = tt.fit_transform(cv_matrix)
         tt_matrix = tt_matrix.toarray()
         vocab = cv.get_feature_names_out()
         tv_matrix = pd.DataFrame(np.round(tt_matrix, 2), columns=vocab)
         tv_matrix
```

Out[12]:

	00	000	05	10	100	11	12	125	13	14	...	zooropa	zornotza	zorro	zubair	zuluaga	zurich	zurici
0	0.0	0.0	0.0	0.00	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1	0.0	0.0	0.0	0.00	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.00	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.00	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.00	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0
...
2220	0.0	0.0	0.0	0.00	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2221	0.0	0.0	0.0	0.00	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2222	0.0	0.0	0.0	0.00	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2223	0.0	0.0	0.0	0.00	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2224	0.0	0.0	0.0	0.04	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0

2225 rows × 31266 columns

Рисунок 3.1.12 - Матриця TF-IDF

Підрахуємо частоту кожного слова.

```
In [13]: df_counts = pd.DataFrame(nltk.FreqDist(words).items(), columns=['words', 'counts'])
         df_counts.set_index('words', inplace=True)
         df_counts
```

Out[13]:

words	counts
quarterly	21
profits	152
us	1807
media	309
giant	152
...	...
191	1
trifling	1
24hours	1
ahhhh	1
lol	1

31338 rows × 1 columns

Рисунок 3.1.13 - Частоти слів

Відсортуємо датафрейм за спаданням частоти.

```
In [14]: df_counts.sort_values(by='counts', ascending=False, inplace=True)
df_counts
```

```
Out[14]:
```

words	counts
said	7252
mr	3004
would	2574
also	2156
people	1968
...	...
braced	1
symbolologist	1
brotherhood	1
illuminati	1
lol	1

31338 rows × 1 columns

Рисунок 3.1.14 - Сортований датафрейм

Зобразимо перші десять найбільш уживаних слів.

```
In [15]: df_counts.head(10)
```

```
Out[15]:
```

words	counts
said	7252
mr	3004
would	2574
also	2156
people	1968
new	1901
us	1807
one	1732
year	1624
could	1495

Рисунок 3.1.15 - Найуживаніші слова

Виведемо метрику для перших десяти елементів

```
In [16]: tv_matrix[df_counts.index[:10].to_list()]
```

```
Out[16]:
```

	said	mr	would	also	people	new	us	one	year	could
0	0.05	0.00	0.00	0.03	0.00	0.00	0.06	0.02	0.00	0.00
1	0.03	0.02	0.00	0.00	0.00	0.04	0.16	0.00	0.02	0.05
2	0.04	0.00	0.02	0.00	0.00	0.00	0.05	0.00	0.00	0.00
3	0.06	0.02	0.02	0.01	0.00	0.00	0.00	0.00	0.07	0.00
4	0.02	0.00	0.00	0.00	0.00	0.00	0.02	0.02	0.02	0.02
...
2220	0.03	0.00	0.02	0.00	0.04	0.04	0.00	0.00	0.00	0.00
2221	0.04	0.00	0.00	0.02	0.09	0.00	0.00	0.02	0.00	0.00
2222	0.00	0.00	0.01	0.00	0.01	0.07	0.04	0.04	0.02	0.02
2223	0.01	0.10	0.00	0.04	0.02	0.00	0.10	0.02	0.03	0.02
2224	0.01	0.00	0.03	0.00	0.09	0.01	0.02	0.04	0.02	0.03

2225 rows × 10 columns

Рисунок 3.1.16 - Метрики для найбільш уживаних слів

3.2 Додаткове завдання 1

Зчитаємо текст.

In [17]:

df_ukr = pd.read_csv('ukr_text.csv')

df_ukr

Out[17]:

		Id	Title	Body
0	http://k.img.com.ua/rss/ua/4013798	Кличко покликав німецьких інвесторів до Києва	Київ - перспективний і відкритий ринок для біз...	
1	http://k.img.com.ua/rss/ua/4001679	З'явилося відео, як байкер почав стріляти у во...	З'явилося відео конфлікту між мотоциклістом...	
2	http://k.img.com.ua/rss/ua/4001390	У центрі Києва посеред вулиці помер чоловік	У Києві на Бессарабській площі вранці в четвер...	
3	http://k.img.com.ua/rss/ua/4001239	Нічний ураган перетворив Хрещатик на смітник	Київ вночі 16 серпня пережив найсильнішу грозу...	
4	http://k.img.com.ua/rss/ua/4001227	Потоп у Києві: столицю наклав ураган з градом	Уночі Київ вкотре накрила негода. Найсильніший...	
...	
1117	http://k.img.com.ua/rss/ua/3194862	Кореспондент: Діамантові руки. Історія успіху...	Київський офіс Класичного ювелірного дому Лобо...	
1118	http://k.img.com.ua/rss/ua/3194633	Кореспондент: Роздача слонів. Янукович щедро ...	20 років тому орден За заслуги – тоді він нази...	
1119	http://k.img.com.ua/rss/ua/3194587	Кореспондент: Рівняння з трьома відомими. Укр...	10 жовтня політичні важковаговики з табору опо...	
1120	http://k.img.com.ua/rss/ua/3194570	Кореспондент: Точка зору. Мета обкрадає кошти...	Добре там, де нас немає. В Ізраїлі ми є, але т...	
1121	http://k.img.com.ua/rss/ua/3194519	Кореспондент: Капітан Океанів. Інтерв'ю зі Св...	Початок розмови Кореспондента зі знаменитим у...	

1122 rows × 3 columns

Рисунок 3.2.1 - Зчитування файлу

Видаємо колонки 'id'.

```
In [18]: df_ukr.drop(['Id'], axis=1, inplace=True)
df_ukr
```

```
Out[18]:
```

		Title	Body
0	Кличко покликав німецьких інвесторів до Києва	Київ - перспективний і відкритий ринок для біз...	
1	З'явилося відео, як байкер почав стріляти у во...	З'явилося відео конфлікту між мотоциклістом...	
2	У центрі Києва посеред вулиці помер чоловік	У Києві на Бессарабській площі вранці в четвер...	
3	Нічний ураган перетворив Хрещатик на смітник	Київ вночі 16 серпня пережив найсильнішу грозу...	
4	Потоп у Києві: столицю наклав ураган з градом	Уночі Київ вкотре накрила негода. Найсильніший...	
...	
1117	Кореспондент: Діамантові руки. Історія успіху...	Київський офіс Класичного ювелірного дому Лобо...	
1118	Кореспондент: Роздача слонів. Янукович щедро ...	20 років тому орден За заслуги – тоді він нази...	
1119	Кореспондент: Рівняння з трьома відомими. Укр...	10 жовтня політичні важковаговики з табору опо...	
1120	Кореспондент: Точка зору. Мета обкрадає кошти...	Добре там, де нас немає. В Ізраїлі ми є, але т...	
1121	Кореспондент: Капітан Океанів. Інтерв'ю зі Св...	Початок розмови Кореспондента зі знаменитим у...	

1122 rows × 2 columns

Рисунок 3.2.2 - Видалення колонок

Видаємо порожні документи, якщо вони є.

```
In [19]: df_ukr = df_ukr[~(df.content.str.strip() == '')]
df_ukr
```

/tmp/ipykernel_6150/2724321335.py:1: UserWarning: Boolean Series key will be reindexed to match DataFrame index.
df_ukr = df_ukr[~(df.content.str.strip() == '')]

```
Out[19]:
```

	Title	Body
0	Кличко покликав німецьких інвесторів до Києва	Київ - перспективний і відкритий ринок для біз...
1	З'явилося відео, як байкер почав стріляти у во...	З'явилося відео конфлікту між мотоциклістом...
2	У центрі Києва посеред вулиці помер чоловік	У Києві на Бессарабській площі вранці в четвер...
3	Нічний ураган перетворив Хрещатик на смітник	Київ вночі 16 серпня пережив найсильнішу грозу...
4	Потоп у Києві: столицю накрив ураган з градом	Уночі Київ вкотре накрила негода. Найсильніший...
...
1117	Кореспондент: Діамантові руки. Історія успіху...	Київський офіс Класичного ювелірного дому Лобо...
1118	Кореспондент: Роздача слонів. Янукович щедро ...	20 років тому орден За заслуги – тоді він нази...
1119	Кореспондент: Рівняння з трьома відомими. Укр...	10 жовтня політичні важковаговики з табору опо...
1120	Кореспондент: Точка зору. Мета обкрадає кошти...	Добре там, де нас немає. В Ізраїлі ми є, але т...
1121	Кореспондент: Капітан Океанів. Інтерв'ю зі Св...	Початок розмови Кореспондента зі знаменитим у...

1122 rows x 2 columns

Рисунок 3.2.3 - Видалення порожніх документів

Визначимо стоп-слова української мови. Завантажимо їх.

```
In [133...] import requests
import ast
url1 = 'https://raw.githubusercontent.com/olegdubetcky/Ukrainian-Stopwords/main/ukrainian'
url2 = 'https://gist.githubusercontent.com/kissarat/bec2bb727c9fb520043a/raw/ba3116872c626...'
r1 = requests.get(url1)
r2 = requests.get(url2)
with open(nltk.data.path[0]+'corpora/stopwords/ukrainian', 'wb') as f:
    f.write(r1.content)
    f.write(r2.content)
with open('stopwords_ua_list.txt') as ff:
    f.write('\n'.join(ast.literal_eval(''.join(ff.readlines()))).encode())
```

Рисунок 3.2.4 - Завантаження стоп-слів

Визначимо стоп-слова.

```
In [134...] import string
import pymorphy2
from nltk.corpus import stopwords
stopwords = stopwords.words("ukrainian")
morph = pymorphy2.MorphAnalyzer(lang='uk')
stop_words = pd.Series(list(set(stopwords+list(string.punctuation))))
stop_words
```

```
Out[134]:
```

0	той
1	одною
2	чом
3	всередині
4	нашим
...	...
2019	чир
2020	поруч
2021	допіру
2022	абиким
2023	тім

Length: 2024, dtype: object

Рисунок 3.2.5 - Визначення стоп-слів

Перетворимо Series на список.

```
In [135...] stop_words = stop_words.to_list()
```

Рисунок 3.2.6 - Перетворення Series на список

Визначимо функцію, що виконує попередню обробку документу. Застосуємо декоратор `pr.vectorize` для того, щоб функція могла працювати з корпусами.

```
In [136... @np.vectorize
def preproc_doc(doc):
    doc = doc.lower()
    doc = re.sub(r'[\s]+', ' ', doc, re.I | re.A)
    doc_words = re.split(f'^a-zA-Z0-9А-ЩБЮЯґЄІіа-щьюяґєіі'+', doc)
    filtered_words = [w for w in doc_words if w not in stop_words]
    doc = ' '.join(filtered_words)
    doc = doc.strip()
    tokens = wpt.tokenize(doc)
    filtered_tokens = [token for token in tokens if token not in stop_words]
    doc = ' '.join(filtered_tokens)
    return doc

df_ukr['content'] = preproc_doc(df_ukr.Body)
df_ukr['content']
```

```
Out[136]: 0    київ перспективний відкритий ринок бізнесу інв...
          1    явилось відео конфлікту мотоциклістом водієм а...
          2    києві бессарабській площі вранці четвер 16 сер...
          3    київ вночі 16 серпня пережив найсильнішу грозу...
          4    уночі київ вкотре накрила негода найсильніший ...
          ...
          1117  київський офіс класичного ювелірного дому лобо...
          1118  20 орден заслуги називався почесний знак прези...
          1119  10 жовтня політичні важковаговики табору опози...
          1120  ізраїлі країна займає перше місце світі надоїв...
          1121  початок розмови кореспондента знаменитим укра...
          Name: content, Length: 1122, dtype: object
```

Рисунок 3.2.7 - Обробка документів

Завантажимо тональний словник української мови.

```
In [137... import csv
url = 'https://raw.githubusercontent.com/lang-uk/tone-dict-uk/master/tone-dict-uk.tsv'
r = requests.get(url)
with open(nltk.data.path[0]+'tone-dict-uk.tsv', 'wb') as f:
    f.write(r.content)

d = {}
with open(nltk.data.path[0]+'tone-dict-uk.tsv', 'r') as csv_file:
    for row in csv.reader(csv_file, delimiter='\t'):
        d[row[0]] = float(row[1])

from nltk.sentiment.vader import SentimentIntensityAnalyzer
SIA = SentimentIntensityAnalyzer()
SIA.lexicon.update(d)
```

Рисунок 3.2.8 - Завантаження тонального словника

Порахуємо оцінку настрою для тексту.

```
In [138... df_ukr['score'] = df_ukr.apply(
    lambda row: SIA.polarity_scores(row.content)["compound"], axis = 1)
df_ukr
```

```
Out[138]:
```

	Title	Body	content	score
0	Кличко покликав німецьких інвесторів до Києва	Київ - перспективний і відкритий ринок для біз...	київ перспективний відкритий ринок бізнесу інв...	0.7506
1	З'явилося відео, як байкер почав стріляти у во...	З'явилося відео конфлікту між мотоциклістом...	явилося відео конфлікту мотоциклістом водієм а...	-0.2500
2	У центрі Києва посеред вулиці помер чоловік	У Києві на Бессарабській площі вранці в четвер...	київ bessarabskii ploshchi vrancti chetver 16 ser...	-0.6124
3	Нічний ураган перетворив Хрещатик на смітник	Київ вночі 16 серпня пережив найсильнішу грозу...	київ вночі 16 серпня пережив найсильнішу грозу...	-0.2500
4	Потоп у Києві: столицю наклав ураган з градом	Уночі Київ вкотре накрила негода. Найсильніший...	уночі київ вкотре накрила негода найсильніший ...	-0.2500
...
1117	Кореспондент: Діамантові руки. Історія успіху...	Київський офіс Класичного ювелірного дому Лобо...	київський офіс класичного ювелірного дому лобо...	0.9584
1118	Кореспондент: Роздача слонів. Янукович щедро ...	20 років тому орден За заслуги – тоді він нази...	20 орден заслуги називався почесний знак прези...	0.9948
1119	Кореспондент: Рівняння з трьома відомими. Укр...	10 жовтня політичні важковаговики з табору опо...	10 жовтня політичні важковаговики табору опози...	0.6124
1120	Кореспондент: Точка зору. Мета обкрадає кошти...	Добре там, де нас немає. В Ізраїлі ми є, але т...	ізраїлі країна займає перше місце світі надоїв...	-0.8402
1121	Кореспондент: Капітан Океанів. Інтерв'ю зі Св...	Початок розмови Кореспондента зі знаменитим у...	початок розмови кореспондента знаменитим укра...	0.9638

1122 rows x 4 columns

Рисунок 3.2.9 - Оцінка настрою

Розділимо текст на матрицю слів.

```
In [139... sentences = [sent.split() for sent in df_ukr.content]
sentences[0][:10]
```

```
Out[139]: ['київ',
            'перспективний',
            'відкритий',
            'ринок',
            'бізнесу',
            'інвестицій',
            'мер',
            'києва',
            'віталій',
            'кличко']
```

Рисунок 3.2.10 - Речення

Виділяємо біграми для всіх документів та створюємо словник.

```
In [140... from gensim.models.phrases import Phrases, Phraser
bigram = Phrases(sentences, min_count=20, threshold=20)
bigram_model = Phraser(bigram)
```

Рисунок 3.2.11 - Модель

Виділяємо біграми для всіх документів та створюємо словник.


```
In [141]: from gensim.corpora import Dictionary
norm_corpus_bigrams = [bigram_model[sent] for sent in sentences]
dictionary = Dictionary(norm_corpus_bigrams)
norm_corpus_bigrams[:1][:10]
```

```
Out[141]: [['київ',
'перспективний',
'відкритий',
'ринок',
'бізнесу',
'інвестицій',
'мер',
'києва',
'віталій',
'кличко',
'заявив',
'виступу',
'дні',
'німецької',
'економіки',
'цьогоріч',
'проходить',
'місті',
'аахені',
'інформує',
'прес_служба',
'четвер',
'20',
'вересня',
'головна',
'тема',
'форуму',
'світова',
'торгівля',
'умовах',
'глобальних',
'змін',
'беруть',
'участь',
'1000',
'чоловік',
'представники',
'50',
'промислово',
'торговельних',
'палат',
'німеччини',
'провідних',
'німецьких',
'компаній',
'здійснюють',
'зовнішньоторговельну',
'діяльність',
'експерти',
'київ',
'підготовлений',
'відкритий',
'співпраці',
'перспективний',
'інвестування',
'кличко',
'зазначив',
'60',
'іноземних',
'інвестицій',
'українську',
'економіку',
'приходять',
'столицю',
'місті',
'введені',
'прозорі',
'механізми',
'управління',
'гарантом',
'інвесторів',
'мер',
'виступає',
'особисто',
'кличко',
'запевнив',
'київ',
'надійним',
'партнером',
```

Рисунок 3.2.12 - Біграми документів

Зменшимо об'єм словника через велику кількість унікальних рідкісних слів. та створюємо модель сумки слів.

```
In [142]: dictionary.filter_extremes(no_below=20, no_above=0.6)
bow_corpus = [dictionary.doc2bow(text) for text in norm_corpus_bigrams]
bow_corpus[:20]
```

```
Out[142]: [(0, 1),
(1, 1),
(2, 1),
(3, 2),
(4, 1),
(5, 1),
(6, 1),
(7, 1),
(8, 2),
(9, 4),
(10, 1),
(11, 1),
(12, 1),
(13, 1),
(14, 1),
(15, 1),
(16, 1),
(17, 1),
(18, 1),
(19, 1),
(20, 1),
(21, 1),
(22, 1),
(23, 1),
(24, 1),
(25, 5),
(26, 1),
(27, 5),
(28, 2),
(29, 8),
(30, 1),
(31, 2),
(32, 2),
(33, 1),
(34, 1),
(35, 1),
(36, 2),
(37, 1),
(38, 2),
(39, 1),
(40, 1),
(41, 1),
(42, 1),
(43, 2),
(44, 1),
(45, 1),
(46, 1),
(47, 1),
(48, 1),
(49, 1),
(50, 1),
(51, 1),
(52, 1),
(53, 1),
(54, 1),
(55, 1),
(56, 1),
(57, 1),
(58, 1),
(59, 1),
(60, 2),
(61, 2),
(62, 1),
(63, 1),
(64, 1),
(65, 1),
(66, 2),
(67, 1),
(68, 1),
(69, 1),
(70, 3),
(71, 1),
(72, 2),
(73, 1),
(74, 1),
(75, 1),
(76, 1),
(77, 2),
(78, 1),
(79, 3),
```

Рисунок 3.2.13 - Сумка слів

Застосуємо приховане семантичне індексування.

```
In [143]: from gensim.models import LsiModel
total_topics = 10
lsi_bow = LsiModel(bow_corpus, id2word=dictionary,
                  num_topics=total_topics,
                  onepass=True, chunksize=10000,
                  power_iters=1000)
```

Рисунок 3.2.14 - Приховане семантичне індексування

Переглянемо основні теми.

```
In [144]: for topic_id, topic in lsi_bow.print_topics(num_topics=10, num_words=20):
          print('Topic #' + str(topic_id+1) + ':')
          print(topic)

Topic #1:
0.667*"1" + 0.457*"2" + 0.346*"0" + 0.247*"3" + 0.175*"4" + 0.139*"5" + 0.095*"7" + 0.087
*"0_0" + 0.081*"6" + 0.073*"10" + 0.066*"україна" + 0.063*"8" + 0.061*"11" + 0.058*"19" +
0.053*"9" + 0.051*"матч" + 0.043*"14" + 0.042*"13" + 0.042*"30" + 0.041*"00"
Topic #2:
0.337*"1" + -0.220*"5" + -0.204*"україни" + -0.160*"6" + -0.156*"10" + -0.151*"україні" +
-0.129*"компанії" + -0.123*"2019" + -0.123*"сша" + -0.115*"україна" + -0.112*"7" + -0.107
*"країни" + -0.105*"30" + -0.104*"8" + 0.101*"0" + -0.098*"9" + -0.093*"00" + -0.090*"груд
ня" + -0.090*"20" + -0.089*"світу"
Topic #3:
-0.344*"1" + 0.300*"00" + 0.228*"5" + 0.221*"10" + 0.218*"30" + 0.203*"6" + 0.161*"11" +
0.154*"14" + 0.149*"2" + 0.142*"0" + -0.129*"компанії" + 0.129*"7" + -0.128*"україні" + -
0.116*"україни" + 0.111*"19" + 0.093*"4" + 0.088*"17" + 0.087*"8" + 0.087*"16" + 0.086*"1
3"
Topic #4:
0.593*"00" + -0.481*"5" + 0.343*"30" + -0.268*"6" + -0.182*"4" + 0.132*"14" + 0.117*"11" +
0.109*"in" + -0.106*"7" + 0.100*"1" + 0.077*"серпня" + 0.075*"україна" + -0.071*"8" + 0.07
0*"день" + -0.065*"3" + 0.060*"0" + 0.058*"the" + 0.056*"бій" + -0.054*"27" + 0.053*"світ
у"
Topic #5:
-0.404*"україна" + 0.263*"00" + -0.252*"матч" + 0.226*"5" + -0.223*"19" + -0.204*"2020" +
0.197*"30" + -0.170*"матчі" + 0.151*"компанії" + -0.147*"90" + 0.136*"google" + -0.136*"ук
раїни" + -0.125*"перемогу" + 0.113*"6" + -0.112*"матчу" + 0.108*"ринку" + -0.104*"українц
і" + -0.100*"онлайн" + -0.094*"поєдинок" + 0.090*"компанія"
Topic #6:
0.346*"області" + -0.305*"google" + 0.290*"інвестицій" + -0.191*"компанії" + -0.181*"рейти
нгу" + -0.161*"ринку" + 0.156*"регіону" + 0.153*"душу_населення" + 0.143*"україни" + 0.138
*"2019" + -0.136*"компанія" + 0.121*"00" + -0.119*"the" + 0.117*"завод" + 0.114*"зростанн
я" + 0.112*"україні" + 0.111*"розвитку" + 0.107*"влади" + 0.106*"підприємств" + 0.104*"рег
іоні"
Topic #7:
-0.232*"області" + -0.216*"google" + -0.213*"інвестицій" + -0.185*"компанії" + 0.184*"сша"
+ 0.184*"грудня" + 0.154*"україни" + -0.139*"рейтингу" + -0.132*"компаній" + -0.130*"матч"
+ 0.129*"заявив" + 0.128*"президент" + -0.123*"ринку" + 0.121*"https_t" + 0.118*"новини_ко
рреспондент" + 0.118*"net_telegram" + 0.117*"me_korrespondentnet" + 0.117*"підписуйтесь_ка
нал" + 0.117*"росії" + -0.114*"україна"
Topic #8:
-0.389*"0" + 0.372*"5" + -0.288*"7" + -0.232*"10" + 0.222*"00" + 0.163*"2019" + -0.148*"9"
+ -0.139*"україна" + -0.137*"8" + 0.130*"матч" + 0.128*"удар" + -0.115*"12" + 0.100*"отрим
ав" + -0.098*"18" + -0.096*"29" + 0.096*"матчу" + 0.094*"2" + 0.093*"матчі" + -0.092*"31"
+ 0.089*"1"
Topic #9:
0.246*"2019" + 0.243*"https_t" + 0.232*"net_telegram" + 0.232*"новини_корреспондент" + 0.2
29*"підписуйтесь_канал" + 0.229*"me_korrespondentnet" + 0.227*"області" + 0.161*"google" +
0.144*"повідомлялося" + 0.125*"компанія" + -0.118*"журналу_корреспондент" + -0.117*"тис" +
-0.117*"україни" + -0.110*"держави" + 0.108*"нагадаємо" + 0.100*"facebook" + -0.100*"5" +
0.097*"грудня" + 0.096*"компанії" + -0.095*"країни"
Topic #10:
0.379*"2" + 0.269*"україна" + -0.253*"1" + 0.241*"україни" + -0.214*"7" + -0.163*"фото" +
-0.151*"новий" + 0.140*"google" + 0.128*"компанії" + -0.124*"9" + -0.113*"євро" + 0.102
*"0" + -0.101*"бій" + 0.100*"компанія" + -0.099*"8" + -0.098*"днів" + 0.097*"заявив" + 0.0
96*"5" + 0.096*"3" + -0.096*"2019"
```

Рисунок 3.2.15 - Основні теми

3.3 Додаткове завдання 2

Зчитуємо тексти обох письменників.

```
In [164... df_poe = []
for text in ['poe.txt', 'poe-2.txt']:
    t = ''.join(open(text).readlines())
    sentences = nltk.tokenize.sent_tokenize(t)
    df_poe.extend(sentences)
df_poe = pd.DataFrame(df_poe, columns=['content'])
df_doyle = []
for text in ['doyle.txt', 'doyle-2.txt']:
    t = ''.join(open(text).readlines())
    sentences = nltk.tokenize.sent_tokenize(t)
    df_doyle.extend(sentences)
df_doyle = pd.DataFrame(df_doyle, columns=['content'])
```

Рисунок 3.3.1 - Зчитування текстів

Визначимо стоп-слова англійської мови.

```
In [165... stop_words = nltk.corpus.stopwords.words('english')
```

Рисунок 3.3.2 - Стоп-слова

Визначимо функцію, що виконує попередню обробку документу. Застосуємо декоратор `np.vectorize` для того, щоб функція могла працювати з корпусами.

```
In [166... @np.vectorize
def preproc_doc(doc):
    doc = re.sub(r'^a-zA-Z\s', '', doc, re.I | re.A)
    doc = doc.lower()
    doc = doc.strip()
    tokens = wpt.tokenize(doc)
    filtered_tokens = [token for token in tokens if token not in stop_words]
    doc = ' '.join(filtered_tokens)
    return doc

df_poe['clean_content'] = preproc_doc(df_poe.content)
df_doyle['clean_content'] = preproc_doc(df_doyle.content)
```

Рисунок 3.3.3 - Обробка документів

Представимо сумку слів для По.

```
In [167... from sklearn.feature_extraction.text import CountVectorizer
cv_poe = CountVectorizer(min_df=0., max_df=1.)
cv_matrix_poe = cv_poe.fit_transform(df_poe.clean_content)
cv_matrix_poe = pd.DataFrame(cv_matrix_poe.toarray(),
                             columns=cv_poe.get_feature_names_out())
cv_matrix_poe
```

```
Out[167]:
```

	abandon	abandoned	abandoning	aberration	ability	able	abound	abovenamed	absence	absent	...	yes
0	0	0	0	0	0	0	0	0	0	0	...	
1	0	0	0	0	0	0	0	0	0	0	...	
2	0	0	0	0	0	0	0	0	0	0	...	
3	0	0	0	0	0	0	0	0	0	0	...	
4	0	0	0	0	0	0	0	0	0	0	...	
...	
1448	0	0	0	0	0	0	0	0	0	0	...	
1449	0	0	0	0	0	0	0	0	0	0	...	
1450	0	0	0	0	0	0	0	0	0	0	...	
1451	0	0	0	0	0	0	0	0	0	0	...	
1452	0	0	0	0	0	0	0	0	0	0	...	

1453 rows × 4209 columns

Рисунок 3.3.4 - Сумка слів для По

Представимо сумку слів для Дойля.

```
In [168... cv_doyle = CountVectorizer(min_df=0., max_df=1.)
cv_matrix_doyle = cv_doyle.fit_transform(df_doyle.clean_content)
cv_matrix_doyle = pd.DataFrame(cv_matrix_doyle.toarray(),
                               columns=cv_doyle.get_feature_names_out())
cv_matrix_doyle
```

Out[168]:

	aback	aban	abandon	abandoned	abandoning	abandons	abbots	aberdeen	abetting	abhor	...	youve
0	0	0	0	0	0	0	0	0	0	0	...	0
1	0	0	0	0	0	0	0	0	0	0	...	0
2	0	0	0	0	0	0	0	0	0	0	...	0
3	0	0	0	0	0	0	0	0	0	0	...	0
4	0	0	0	0	0	0	0	0	0	0	...	0
...
10731	0	0	0	0	0	0	0	0	0	0	...	0
10732	0	0	0	0	0	0	0	0	0	0	...	0
10733	0	0	0	0	0	0	0	0	0	0	...	0
10734	0	0	0	0	0	0	0	0	0	0	...	0
10735	0	0	0	0	0	0	0	0	0	0	...	0

10736 rows × 10656 columns

Рисунок 3.3.5 - Сумка слів для Дойля

Визначимо настрої кожного речення за допомогою TextBlob. Зробимо це для По.

```
In [169... from textblob import TextBlob
df_poe['score'] = df_poe.apply(
    lambda row: TextBlob(row.clean_content).sentiment.polarity, axis = 1)
df_poe
```

Out[169]:

	content	clean_content	score
0	1843\n\n...	goldbug edgar allan poe goldbug ho	0.0000
1	what ho!	ho	0.0000
2	this fellow is dancing mad!	fellow dancing mad	-0.6250
3	He hath been bitten by the Tarantula.	hath bitten tarantula	0.0000
4	All in the Wrong.	wrong	-0.5000
...
1448	But he is a\ngood creature after all.	good creature	0.7000
1449	I like him especially for one master stroke\no...	like especially one master stroke cant attaine...	0.0000
1450	I mean\nthe way he has 'de nier ce qui est, et...	mean way de nier ce qui est et dexpliquer ce q...	-0.3125
1451	""\n\n * Rousseau, Nouvelle Heloise.	rousseau nouvelle heloise	0.0000
1452	-THE END-\n.	end	0.0000

1453 rows × 3 columns

Рисунок 3.3.6 - Оцінка настрою для По

Зробимо це для Дойля.

```
In [170]: df_doyle['score'] = df_doyle.apply(
          lambda row: TextBlob(row.clean_content).sentiment.polarity, axis = 1)
          df_doyle
```

```
Out[170]:
```

	content	clean_content	score
0	[obi/Doyle/hound.of.baskervilles.txt]\n\n ...	obidoylehoundofbaskervilles.txt chapter mr sher...	-0.300000
1	I stood upon the\nhearth-rug and picked up the...	stood upon hearthrug picked stick visitor left...	-0.200000
2	It was a fine, thick piece of wood,\nbulbous-h...	fine thick piece wood bulbousheaded sort known...	0.058333
3	Just under the head was a broad silver band ne...	head broad silver band nearly inch across	0.081250
4	"To James Mortimer, M.R.C.S., from his friends...	james mortimer mrCs friends cch engraved upon ...	0.000000
...
10731	Mr. Rucastle survived, but\nwas always a broke...	mr rucastle survived always broken man kept al...	-0.150000
10732	They still live with their old servants, who\n...	still live old servants probably know much ruc...	-0.062727
10733	Mr. Fowler and Miss Rucastle were\nmarried, by...	mr fowler miss rucastle married special licens...	0.303571
10734	As to Miss Violet Hunter, my friend\nHolmes, r...	miss violet hunter friend holmes rather disapp...	-0.050000
10735	.	.	0.000000

10736 rows × 3 columns

Рисунок 3.3.7 - Оцінка настрою для Дойля

Визначимо оцінки настрою для кожного автора.

```
In [172]: print('Poe: ', df_poe.score.mean())
          print('Doyle: ', df_doyle.score.mean())

Poe:  0.036458169953032804
Doyle: 0.03274252363362101
```

Рисунок 3.3.8 - Оцінки настрою

Дойль похмуріший за По.

ДОДАТОК А ТЕКСТИ ПРОГРАМНОГО КОДУ

Тексти програмного коду
(Найменування програми (документа))

Жорсткий диск

(Вид носія даних)

(Обсяг програми (документа), арк.)

Студента групи ІІІ-113 курсу
Панченка С. В


```

import numpy as np
import pandas as pd
import nltk
import re
df = pd.read_csv('bbc-news-data.csv', sep='\t')
df
df.drop(['filename', 'title', 'category'], axis=1, inplace=True)
df
df = df[~(df.content.str.strip() == '')]
df
wpt = nltk.WordPunctTokenizer()
stop_words = nltk.corpus.stopwords.words('english')
@np.vectorize
def preproc_doc(doc):
    doc = re.sub(r'[^a-zA-Z\s]', '', doc, re.I | re.A)
    doc = doc.lower()
    doc = doc.strip()
    tokens = wpt.tokenize(doc)
    filtered_tokens = [token for token in tokens if token not in stop_words]
    doc = ' '.join(filtered_tokens)
    return doc
p_corpus = preproc_doc(df.content)
p_corpus
words = []
for doc in p_corpus:
    words.extend(doc.split(' '))
df_words = pd.DataFrame(set(words))
df_words.columns = ['words']
df_words.head(10)
df_words['ps'] = [tag for _, tag in nltk.pos_tag(df_words.words)]
df_words
from nltk.tag import map_tag
df_words['sps'] = [map_tag('en-ptb', 'universal', tag) for tag in df_words.ps]
df_words
abbr = {'NOUN': 'n', 'VERB': 'v', 'ADJ': 'a', 'ADV': 'r'}
def to_abbr(el):
    res = abbr.get(el)
    if res is not None:
        return res
    return ''
df_words['abbr'] = [to_abbr(x) for x in df_words.sps]
df_words
from nltk.stem import WordNetLemmatizer
wlem = WordNetLemmatizer()

```

```

def my_lem(word, abbr):
    if abbr == '':
        return wlem.lemmatize(word)
    return wlem.lemmatize(word, pos=abbr)
df_words['lemms'] = [my_lem(row[1]['words'], row[1]['abbr'])
for row in df_words.loc[:, ['words', 'abbr']].iterrows()]
df_words
from sklearn.feature_extraction.text import CountVectorizer
cv = CountVectorizer(min_df=0., max_df=1.)
cv_matrix = cv.fit_transform(p_corpus)
cv_matrix = pd.DataFrame(cv_matrix.toarray(),
columns=cv.get_feature_names_out())
cv_matrix.to_csv('bag_of_words.csv')
from sklearn.feature_extraction.text import TfidfTransformer
tt = TfidfTransformer(norm='l2', use_idf=True)
tt_matrix = tt.fit_transform(cv_matrix)
tt_matrix = tt_matrix.toarray()
vocab = cv.get_feature_names_out()
tv_matrix = pd.DataFrame(np.round(tt_matrix, 2), columns=vocab)
tv_matrix
df_counts = pd.DataFrame(nltk.FreqDist(words).items(), columns=['words',
'counts'])
df_counts.set_index('words', inplace=True)
df_counts
df_counts.sort_values(by='counts', ascending=False, inplace=True)
df_counts
df_counts.head(10)
tv_matrix[df_counts.index[:10].to_list()]
df_ukr = pd.read_csv('ukr_text.csv')
df_ukr
df_ukr.drop(['Id'], axis=1, inplace=True)
df_ukr
df_ukr = df_ukr[~(df.content.str.strip() == '')]
df_ukr
import requests
import ast
url1 = 'https://raw.githubusercontent.com/olegdubetcky/Ukrainian-Stopwords/main/
ukrainian'
url2 = 'https://gist.githubusercontent.com/kissarat/bec2bb727c9fb520043a/raw/
ba3116872c6261ceaa0a9f4db616c742f7d3cba0/ukrainian-stopwords.txt'
r1 = requests.get(url1)
r2 = requests.get(url2)
with open(nltk.data.path[0]+'corpora/stopwords/ukrainian', 'wb') as f:
    f.write(r1.content)
    f.write(r2.content)
with open('stopwords_ua_list.txt') as ff:

```

```

f.write('\n'.join(ast.literal_eval(''.join(ff.readlines()))).encode())
import string
import pymorphy2
from nltk.corpus import stopwords
stopwords = stopwords.words("ukrainian")
morph = pymorphy2.MorphAnalyzer(lang='uk')
stop_words = pd.Series(list(set(stopwords+list(string.punctuation))))
stop_words
stop_words = stop_words.to_list()
@np.vectorize
def preproc_doc(doc):
    doc = doc.lower()
    doc = re.sub(r'[\s]+', ' ', doc, re.I | re.A)
    doc_words = re.split(f'^a-zA-Z0-9A-ЩЮЯҐЄІİа-щьюяґєіі+', doc)
    filtered_words = [w for w in doc_words if w not in stop_words]
    doc = ' '.join(filtered_words)
    doc = doc.strip()
    tokens = wpt.tokenize(doc)
    filtered_tokens = [token for token in tokens if token not in stop_words]
    doc = ' '.join(filtered_tokens)
    return doc
df_ukr['content'] = preproc_doc(df_ukr.Body)
df_ukr['content']
import csv
url = 'https://raw.githubusercontent.com/lang-uk/tone-dict-uk/master/tone-dict-uk.tsv'
r = requests.get(url)
with open(nltk.data.path[0]+'tone-dict-uk.tsv', 'wb') as f:
    f.write(r.content)
d = {}
with open(nltk.data.path[0]+'tone-dict-uk.tsv', 'r') as csv_file:
    for row in csv.reader(csv_file, delimiter='\t'):
        d[row[0]] = float(row[1])
from nltk.sentiment.vader import SentimentIntensityAnalyzer
SIA = SentimentIntensityAnalyzer()
SIA.lexicon.update(d)
df_ukr['score'] = df_ukr.apply(
    lambda row: SIA.polarity_scores(row.content)["compound"], axis = 1)
df_ukr
sentences = [sent.split() for sent in df_ukr.content]
sentences[0][:10]
from gensim.models.phrases import Phrases, Phraser
bigram = Phrases(sentences, min_count=20, threshold=20)
bigram_model = Phraser(bigram)
from gensim.corpora import Dictionary
norm_corpus_bigrams = [bigram_model[sent] for sent in sentences]

```

```

dictionary = Dictionary(norm_corpus_bigrams)
norm_corpus_bigrams[:1][:10]
dictionary.filter_extremes(no_below=20, no_above=0.6)
bow_corpus = [dictionary.doc2bow(text) for text in norm_corpus_bigrams]
bow_corpus[:20]
from gensim.models import LsiModel
total_topics = 10
lsi_bow = LsiModel(bow_corpus, id2word=dictionary,
num_topics=total_topics,
onepass=True, chunksize=10000,
power_iters=1000)
for topic_id, topic in lsi_bow.print_topics(num_topics=10, num_words=20):
print('Topic #' + str(topic_id + 1) + ':')
print(topic)
df_poe = []
for text in ['poe.txt', 'poe-2.txt']:
t = ''.join(open(text).readlines())
sentences = nltk.tokenize.sent_tokenize(t)
df_poe.extend(sentences)
df_poe = pd.DataFrame(df_poe, columns=['content'])
df_doyle = []
for text in ['doyle.txt', 'doyle-2.txt']:
t = ''.join(open(text).readlines())
sentences = nltk.tokenize.sent_tokenize(t)
df_doyle.extend(sentences)
df_doyle = pd.DataFrame(df_doyle, columns=['content'])
stop_words = nltk.corpus.stopwords.words('english')
@np.vectorize
def preproc_doc(doc):
doc = re.sub(r'^a-zA-Z\s', '', doc, re.I | re.A)
doc = doc.lower()
doc = doc.strip()
tokens = wpt.tokenize(doc)
filtered_tokens = [token for token in tokens if token not in stop_words]
doc = ' '.join(filtered_tokens)
return doc
df_poe['clean_content'] = preproc_doc(df_poe.content)
df_doyle['clean_content'] = preproc_doc(df_doyle.content)
from sklearn.feature_extraction.text import CountVectorizer
cv_poe = CountVectorizer(min_df=0., max_df=1.)
cv_matrix_poe = cv_poe.fit_transform(df_poe.clean_content)
cv_matrix_poe = pd.DataFrame(cv_matrix_poe.toarray(),
columns=cv_poe.get_feature_names_out())
cv_matrix_poe
cv_doyle = CountVectorizer(min_df=0., max_df=1.)
cv_matrix_doyle = cv_doyle.fit_transform(df_doyle.clean_content)

```

```
cv_matrix_doyle = pd.DataFrame(cv_matrix_doyle.toarray(),
                                columns=cv_doyle.get_feature_names_out())
cv_matrix_doyle
from textblob import TextBlob
df_poe['score'] = df_poe.apply(
    lambda row: TextBlob(row.clean_content).sentiment.polarity, axis = 1)
df_poe
df_doyle['score'] = df_doyle.apply(
    lambda row: TextBlob(row.clean_content).sentiment.polarity, axis = 1)
df_doyle
print('Poe: ', df_poe.score.mean())
print('Doyle: ', df_doyle.score.mean())
```