Обробка текстів засобами мови Python

Мова програмування

Python – високорівнева, інтерпретована, об’єктно-орієнтована мова програмування.

Python надає набір потужних інструментів опрацювання природної мови:

- різноманітні модулі
- ресурси платформи NLTK (Natural Language Tool Kit)

NLTK

NLTK є провідною платформою для створення Python-програм для роботи з текстами природньою мовою. Вона надає можливість використання:
більш ніж 50 корпусів і лексичних ресурсів
бібліотек для обробки тексту з метою їх класифікації, токенізації, стемінгу, тегування, синтаксичного аналізу та семантичного пояснення.

Класичні задачі обробки текстів

- Базова статистика (визначення кількості абзаців, речень, слів, символів із пробілами та без пробілів)
- Токенізація (поділ тексту на токени – слова/речення)
- Видалення стоп-слів
- Видалення знаків пунктуації
- Визначення частотності слів, побудова частотних словників

Уміння розв'язувати базові задачі надає можливість:

- ➢ розробляти інші види словників
- ➢ проектувати мовні експерименти, проводити дослідження в галузях лексичної семантики, психолінгвістики, морфології та інших
- ➢ створювати прикладні програми, орієнтовані на опрацювання природної мови:
  - ✓ визначення авторства тексту
  - ✓ визначення унікальності тексту
  - ✓ перевірки орфографії та ін.

Програми перевірки орфографії

Програми перевірки орфографії

Програма перевірки орфографії — це інструмент для виправлення помилок. Він є доступним у текстових процесорах, програмах електронної пошти, мобільних телефонах, месенджерах та багатьох інших програмах. Програма сигналізує про неправильно введене слово, виправляє його під час введення, дає можливість шукати слова з помилками в усьому документі.

Створення професійної програми перевірки орфографії є складним процесом, який передбачає використання потужного математичного апарату, крос-лінгвістичного аналізу, засобів нейронних мереж

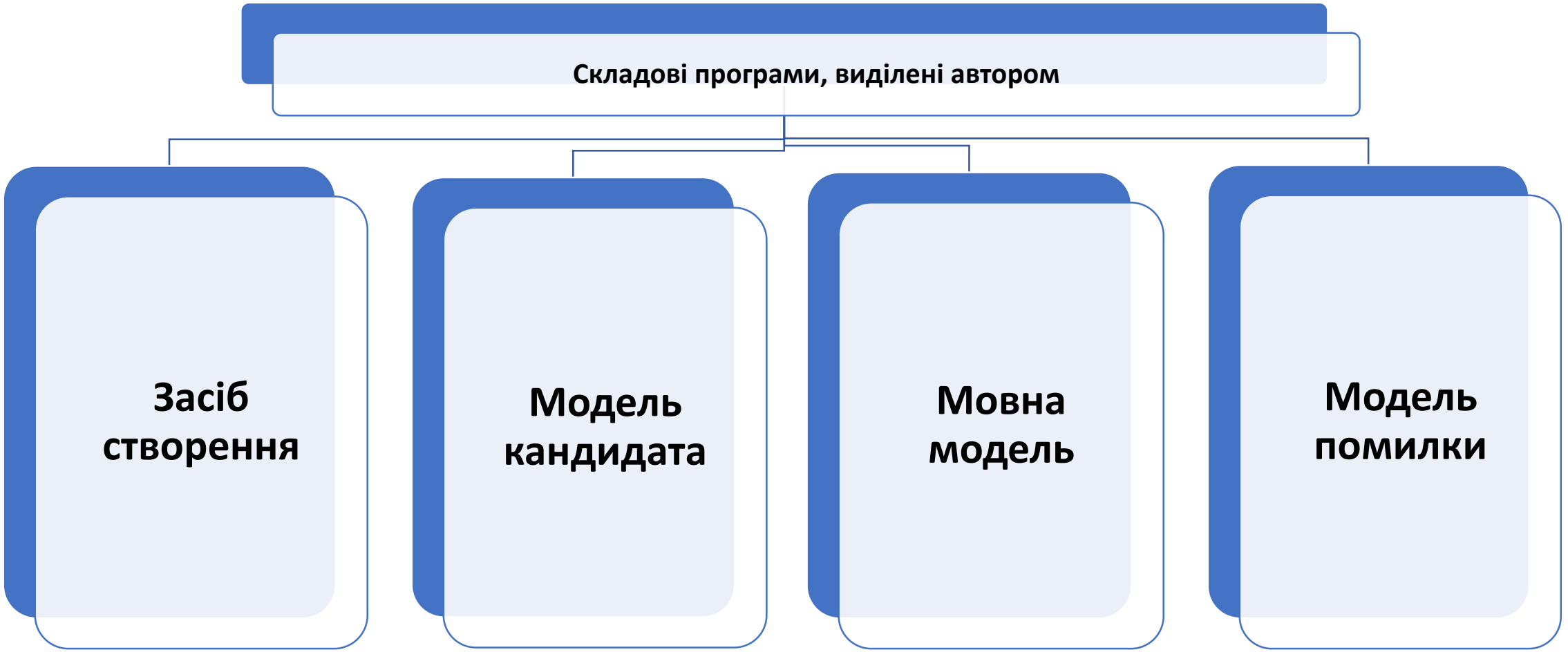
Програма перевірки орфографії Пітера Норвіга

Пітер Норвиг (англ. Peter Norvig) — *американський вчений в галузі IT. Працює директором з досліджень у корпорації Google. Член Ради Американської асоціації з розвитку штучного інтелекту, автор книги*

Штучний інтелект: Сучасний підхід.

Програма перевірки орфографії, Пітера Норвіга має такі переваги як простота, зрозумілість, короткий код, точність (75%), швидкість роботи (опрацювання 35-40 слів за секунду).

Програма перевірки орфографії Пітера Норвіга



Програма перевірки орфографії Пітера Норвіга

Засіб створення

мова програмування (вибрано Python)

Програма перевірки орфографії Пітера Норвіга

Модель кандидата

множина всіх можливих виправлень для заданого слова:

- видалення (видалити одну букву),
- транспозиція (переставити дві сусідні букви),
- заміна (замінити одну букву іншою),
- вставка (вставити букву).

Відповідна функція програми повертає множину рядків (рядок може бути словом або ні), які отримуються за допомогою одного простого виправлення, тобто відстань редагування дорівнює один

Програма перевірки орфографії Пітера Норвіга

Мовна модель

забезпечення можливості оцінки імовірності слова для виправлення за допомогою підрахунку його частотності в заданому текстовому файлі. Бажано, щоб файл містив близько мільйона слів.

Для досягнення цієї мети автор пропонує використовувати великі масиви текстових даних, такі як книги проекту Гутенберг, списки слів Вікісловника, Британський національний корпус.

Програма перевірки орфографії Пітера Норвіга

Модель помилки

створення непорожньої множини слів-кандидатів для виправлення у порядку пріоритету:

1. задане слово, якщо воно відоме; інакше

2. список відомих слів із відстанню редагування один, якщо вони є; інакше

3.список відомих слів із відстанню редагування два, якщо вони є; інакше

4. задане слово, навіть якщо воно невідоме.

Програма перевірки орфографії текстів українською мовою

Поставлені задачі

1. Перетворити функцію редагування для роботи з символами українського алфавіту.
2. Для покращення статистичного аналізу результатів роботи програми змінити функцію створення множини слів-кандидатів, таким чином, щоб вона повертала значення «None» у разі, якщо варіантів виправлення для слова не знайдено.
3. Дібрати лексичний матеріал для забезпечення мовної моделі та визначити засоби його опрацювання.
4. Побудувати регулярний вираз для виокремлення слів із заданого тексту таким чином, щоб українські слова правильно опрацьовувалися, наприклад, слова з апострофом.

Результат роботи функції edits1() – одне просте виправлення

```
>>> edits1('житя')
{'житян', ', 'жшитя', 'жиштя', 'дитя', 'жртя', 'єитя', 'ждтя', 'итя',
'жмтя', 'фжитя', 'жцтя', 'жится', 'житєя', 'жиітя', 'юитя', 'жбтя', ',
'бжитя', 'житяь', 'житз', 'жихтя', 'житяа', 'жлтя', 'житяд', 'жйитя',
'житящ', 'житл', 'житс', 'жятя', 'житья', 'житкя', 'житю', 'яжитя',
'ьжитя', 'жмитя', 'житяи', 'житїя', 'жтия', 'ситя', 'житяй', 'ґитя',
'життя', 'житшя', 'жстя', 'жибя', 'йжитя', 'жьитя', 'жїитя', 'житр',
'жичтя', 'жптя', 'жчтя', 'житвя', 'жити', 'ґжитя', 'жюитя', 'житґ', 'жишя',
'житі', 'єжитя', ', 'жиця', 'житяі', 'жиія', 'ижтя', 'житоя', 'житбя',
'жітя', 'жинтя', 'жиятя', 'жатя', 'житяя', 'житп', 'жить', 'житяб',
'житюя', 'житш', 'жуитя', 'жотя', 'жито', 'жфтя', 'жидтя', 'жйтя', 'жигтя',
'яитя', 'жїтя', 'жзитя', 'житяк', 'зжитя', 'житфя', 'житу', 'аитя', ',
'жизтя', 'жґтя', 'жоитя', 'жистя', 'цитя', 'жицтя', 'жиїтя', 'жщитя',
'жиитя', 'ждитя', 'жщтя', 'житн', 'жаитя', 'жит', 'ажитя', 'чжитя',
'житяї'}
```

Кількість елементів - 297

Результат роботи функції known()

```
>>> known(edits1('житя'))
{'дитя', 'життя', 'жито'}
```

Результат роботи функцій edits2() і known()

```
>>> len(set(edits2('житя')))
39332
>>> known(edits2('житя'))
{'житло', 'пити', 'жива', 'дитям', 'житті',
'жито', 'дитя', 'пиття', 'життя'}
```

Результат роботи функцій candidates() і correction()

```
>>> candidates('житя')
{'життя', 'дитя', 'жито'}
```

Для рядка 'абабаґаламаґа' результатом буде значення [None], оскільки такого слова в словнику немає:

```
>>> candidates('абабаґаламаґа')
[None]
```

```
>>> correction('житя')
'життя'
```

Реалізація мовної моделі

Релевантним ресурсом для створення словника стали матеріали «Браунського корпусу української мови» (https://github.com/brown-uk/corpus).

Був побудований регулярний вираз для виокремлення слів із тексту українською мовою

Проблема апострофа

- "doctor’s" -> "doctor "
- "обов'язковий" -> "обов"

Тестування програми перевірки орфографії

В якості матеріалу для перевірки орфографії та аналізу результатів використано тексти, що містять субтитри українською мовою. Ці тексти створено в рамках проекту **«To Be Announced» – волонтерської програми**, учасники якої здійснюють переклад англомовних серіалів українською мовою та створюють субтитри на основі цих перекладів. Мета втілення проекту – збільшити обсяг україномовного контенту та зробити його більш доступним для усіх верств населення. Тексти, загальним обсягом 289795 слів, включають матеріали різних функціональних стилів, а також велику кількість субстандартної лексики та власних назв.

Результати роботи програми на основі тексту субтитрів до серіалу «13 Reasons Why»

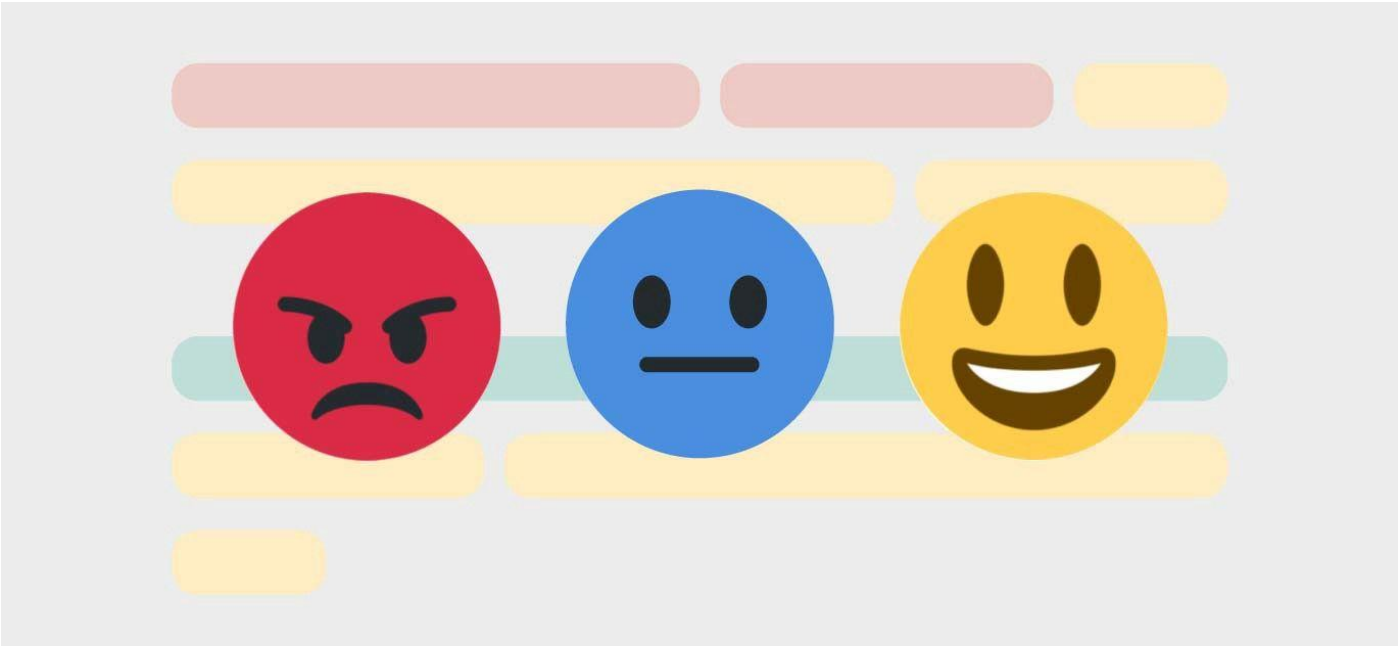
Загальний обсяг тексту	23721 слово
Час виконання програми	1167.1577906608582 секунд або 19 хвилин 45 секунд
Загальна кількість виправлень	2649
Кількість унікальних помилок	1403
Кількість повторів помилок	1246
Кількість справжніх помилок	30
Кількість помилкових виправлень (наприклад, бредлі --- брехні, відповіси --- відповісти, бісові --- лісові)	1373
Кількість слів, для яких запропоновано виправлення	2303
Кількість слів, для яких не запропоновано виправлення	346

Висновки

Помилки складають 11,16% від загального обсягу тексту, з яких 1403 одиниці є унікальними (52,96% від загальної кількості помилок), а 1246 (47,03%) є дублікатами. Лише 30 слів (0,1265%) є справжніми помилками, а інші слова відсутні у створеному словнику для перевірки орфографії. Це зумовлено трьома ключовими факторами:

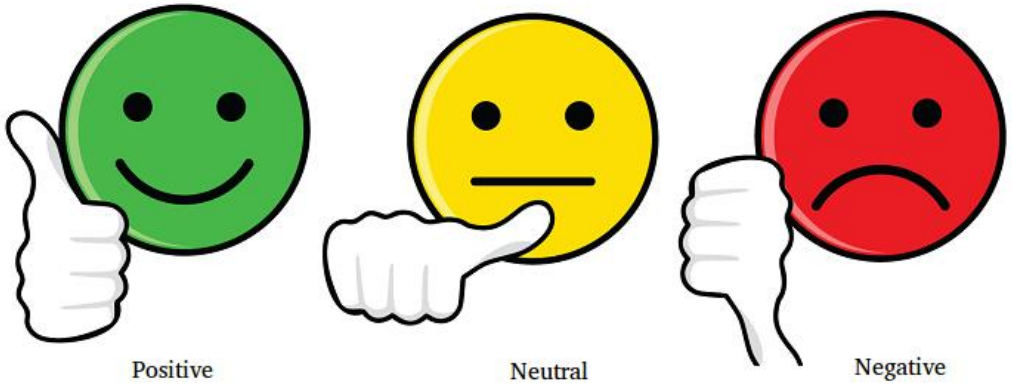
1. українська мова – це синтетична мова і форми слів утворюються за допомогою префіксів, суфіксів та закінчень. Тому, для правильної роботи програми словник повинен містити всі форми слів. Однак, обсяг такого словника буде занадто великим і робота з ним буде тривалою та непродуктивною.
2. жанр – текст містить багато субстандартної лексики, переважно сленгу (**чуваче, пресуха**), який не включено в словник перевірки орфографії;
3. 3. більшість власних назв, що зустрічаються в тексті (**Ханна, Джастін**) та їхні словоформи відсутні в словнику для перевірки орфографії

Аналіз тональності та об’єктивності тексту (Sentiment analysis)



Аналіз тональності тексту

- ✓ **Sentiment analysis** – один із напрямів комп’ютерного опрацювання текстів (natural language processing (NLP))
- ✓ **Sentiment analysis** – метод, який використовується для того, щоб визначити чи є текстове повідомлення емоційно позитивним, негативним чи нейтральним
- ✓ **Sentiment analysis** – є однією зобластей досліджень в комп’ютерних науках, яка найшвидше розвивається, що пов’язано із появою суб’єктивних текстів в інтернеті

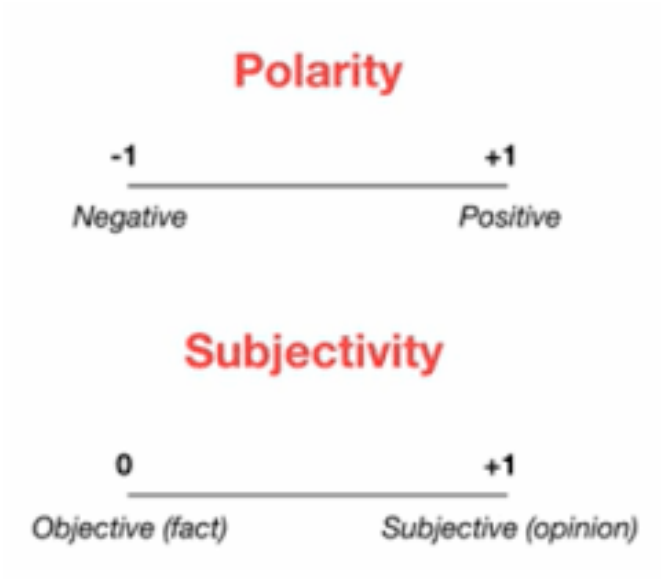


Аналіз суб’єктивності тексту

Іншим напрямком сентимент-аналізу є визначення рівня **суб’єктивності тексту**.

Тобто, текст може оцінюватися як об’єктивний або суб’єктивний

Вимірювання тональності та суб’єктивності



Полярність (Polarity):

- -1 — дуже негативно
- 0 — нейтрально
- 1 — дуже позитивно

Суб’єктивність (Subjectivity):

- 0 — це факт
- +1 — багато думок

Практичне застосування (бізнес)

Sentiment analysis часто виконується на текстових даних, щоб допомогти компаніям відстежувати ставлення до бренду або продукту у відгуках клієнтів та дописах у соціальних мережах і краще розуміти їхні потреби.

- прийняття правильних рішень у ситуаціях електронного маркетингу
- покращення якості послуг мережі мобільного зв’язку
- виявлення побічних реакцій на ліки із текстів, опублікованих пацієнтами у Twitter
- визначення зацікавленості кандидата на посаду в (використовується в рекрутингових компаніях)

Практичне застосування (бізнес)

- *Sentiment Analysis on Airline Tweets*
- **Twitter US Airline Sentiment**
- *https://www.kaggle.com/datasets/crowdflower/twitter-airline-sentiment*
- *https://raw.githubusercontent.com/ruchitgandhi/Twitter-Airline-Sentiment-Analysis/master/Tweets.csv*

Практичне застосування (соціальна та політична сфера)

- прогнозування результатів виборів
- визначення впливу погоди на людські емоції
- моніторинг стихійних лих

Практичне застосування (освіта)

- визначення настроїв студентів у середовищі навчальних платформ
- визначення задоволеності студентів масовими відкритими онлайн курсами
- прогнозування академічної успішності студентів

Бібліотека TextBlob

Бібліотека TextBlob містить модуль sentiment, який має два значення:

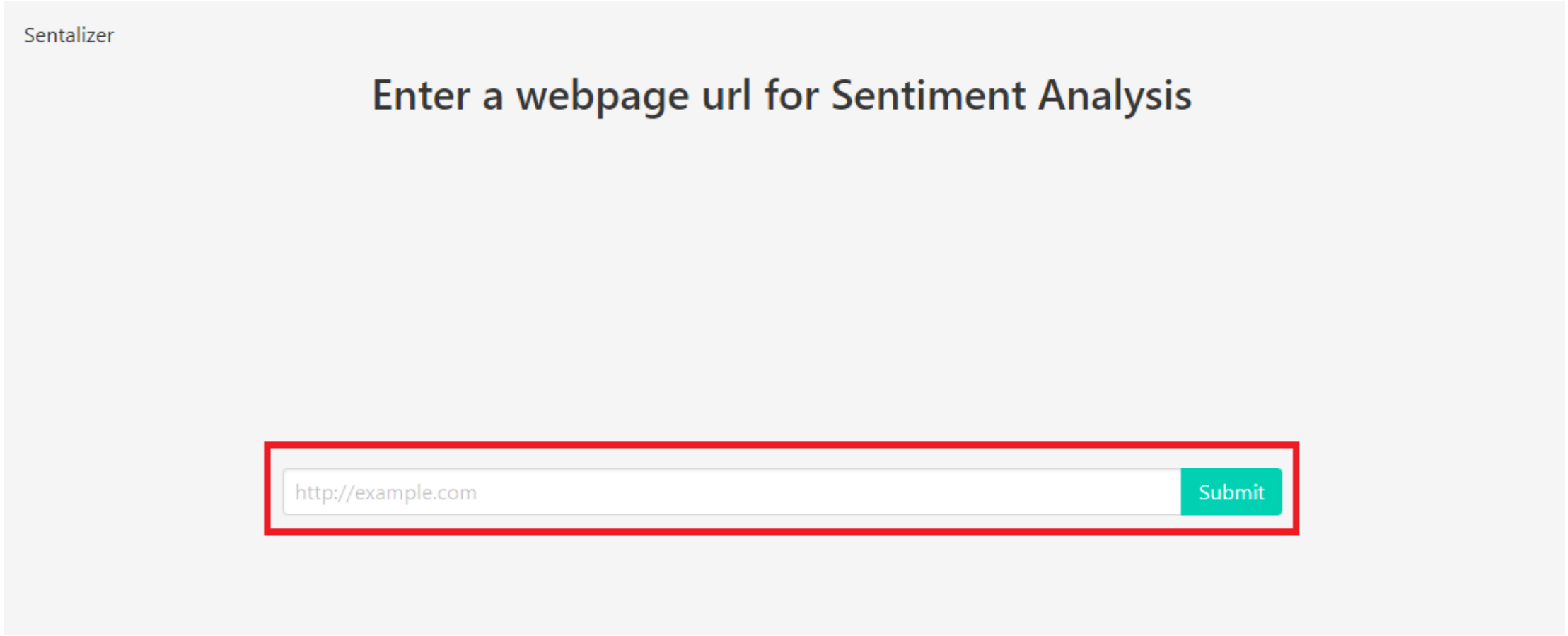
- polarity (полярність)
- subjectivity (суб’єктивність)

Застосування:

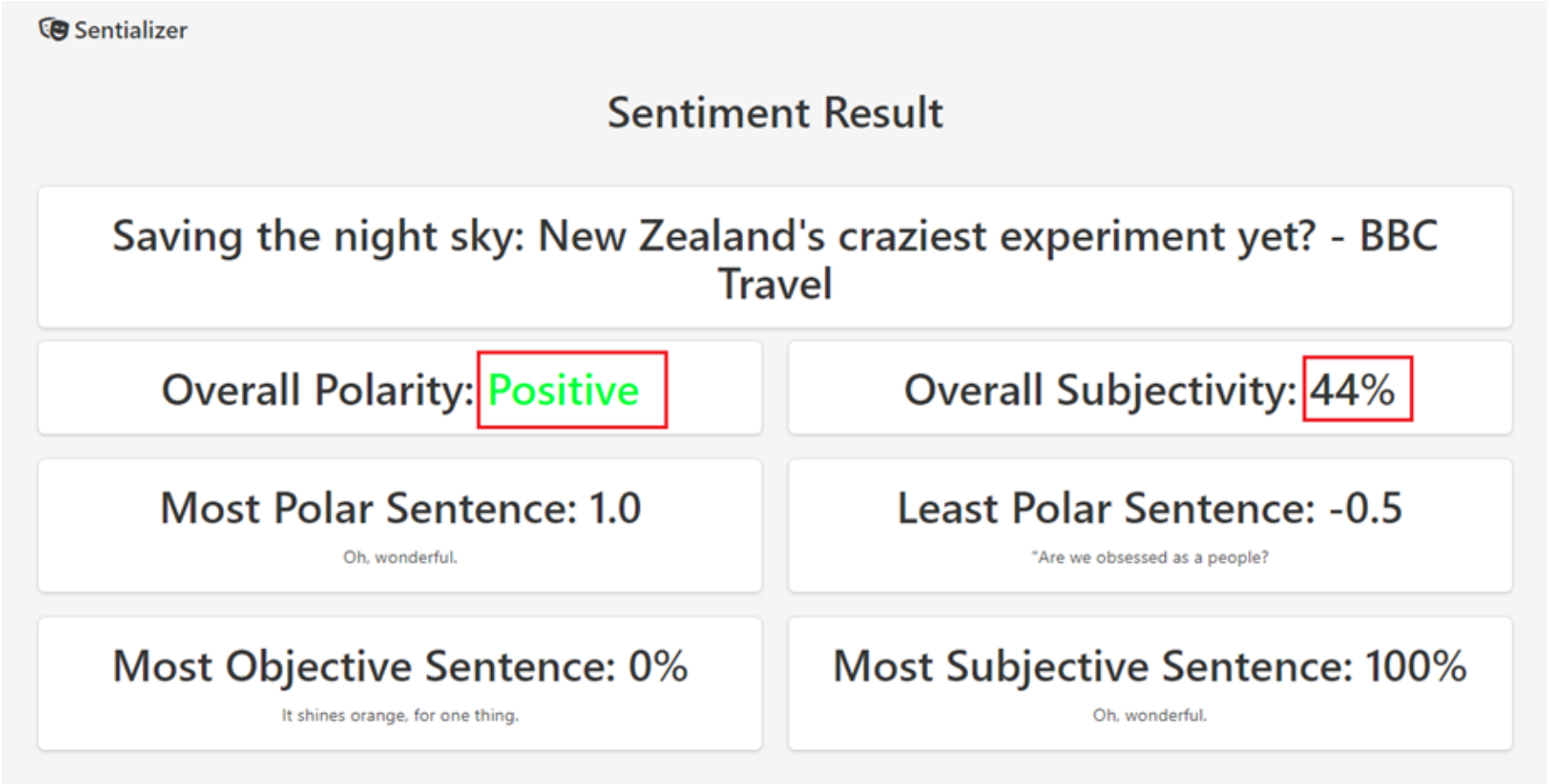
```
from textblob import TextBlob
blob = TextBlob("I really enjoy programming in Python")
result = blob.sentiment
print(result)
```

```
Sentiment(polarity=0.4, subjectivity=0.5)
```

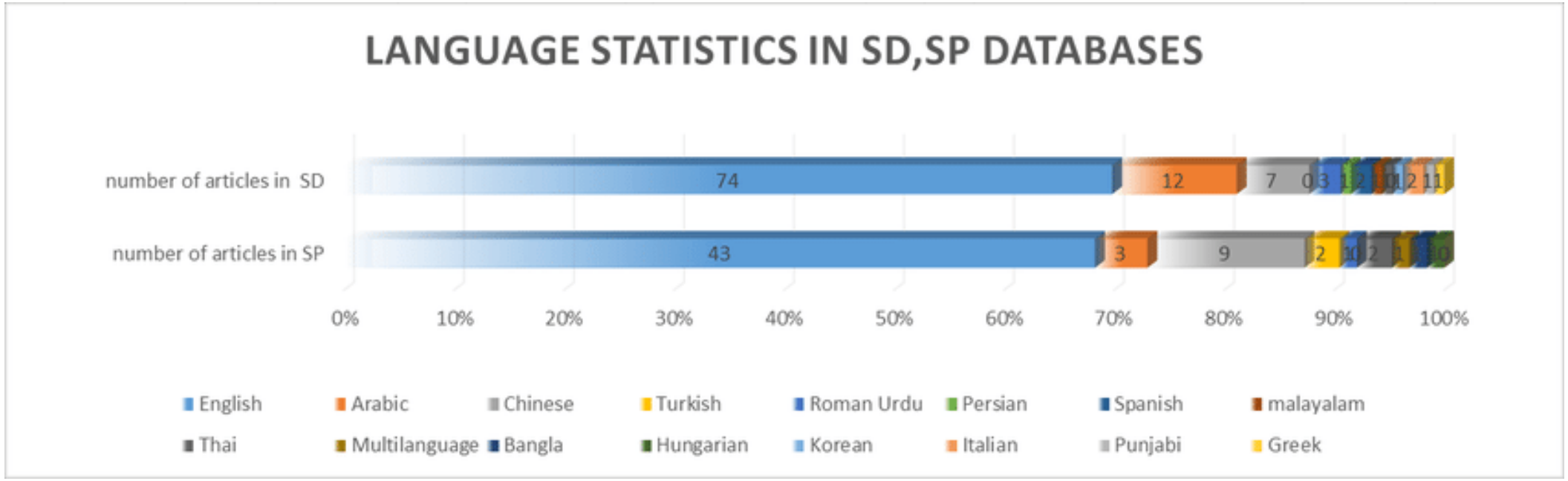
Веб-додаток SENTIALIZER. Інтерфейс для введення даних



Веб-додаток SENTIALIZER. Інтерфейс для виведення результатів



Програми sentiment analysis опрацьовують переважно англійську мову



Створення програми sentiment analysis із використанням машинного перекладу

Googletrans – це безкоштовна Python-бібліотека, у якій реалізовано підтримка Google Translate API

Приклад:

```
from googletrans import Translator

translator = Translator()

translator.translate('안녕하세요.')
```

Результат :

```
<Translated src=ko dest=en text=Good evening. pronunciation=Good evening.>
```

Створення програми sentiment analysis із використанням машинного перекладу

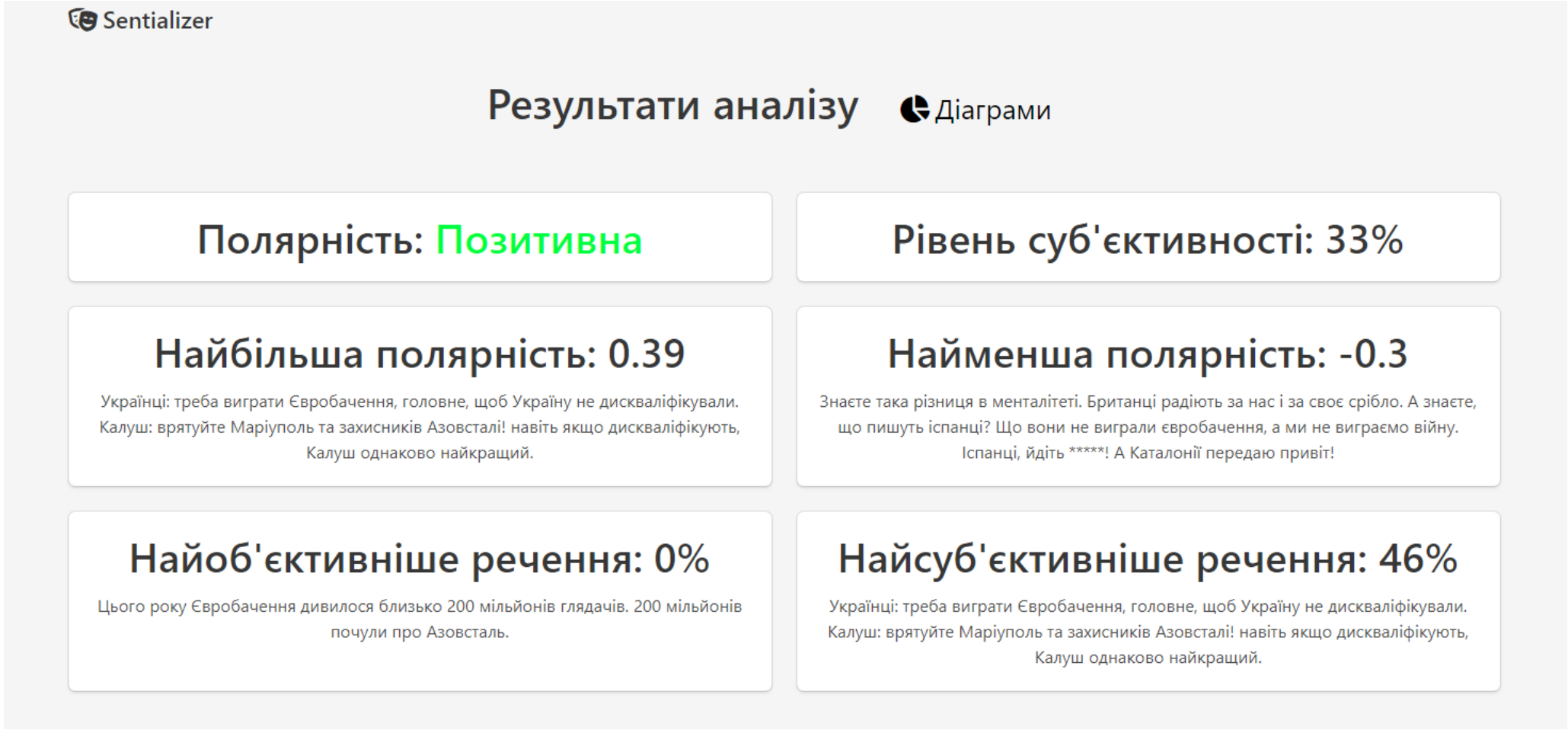
Функція аналізу тексту користувача

```
input_text = request.form['usertext']

detected_language = translator.detect(input_text)

detected_language = detected_language.lang

if detected_language == 'en':

    pass

else:

    inp_list = tokenize.sent_tokenize(input_text)

    inp_list_translated = []

    for sentence in inp_list:

        sentence = translator.translate(sentence, dest='en')

        sentence_translated = sentence.text

     inp_list_translated.append(sentence_translated)
```

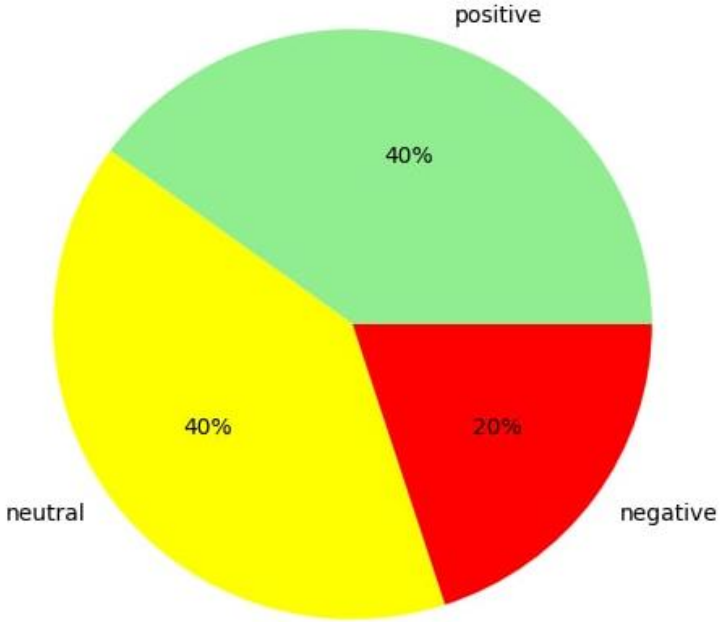
Веб-додаток SENTIALIZER для мультилінгвального аналізу

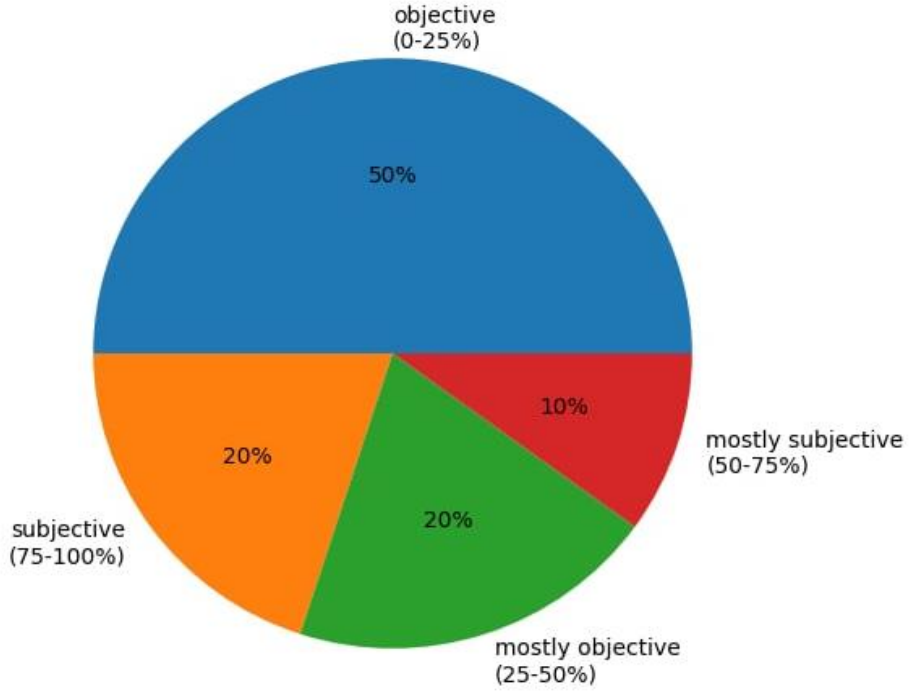
https://sentializer.pythonanywhere.com/

Результати тестування. Твіти про Євробачення

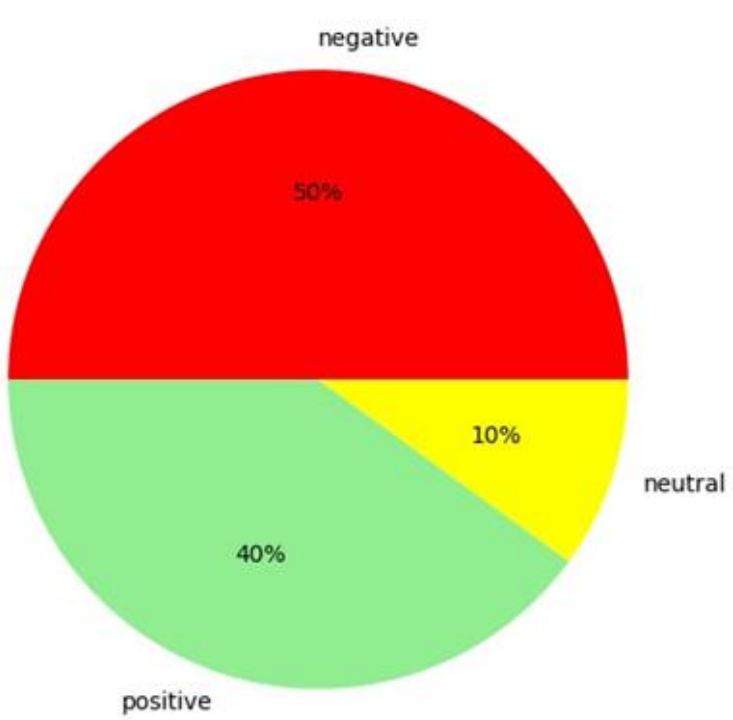


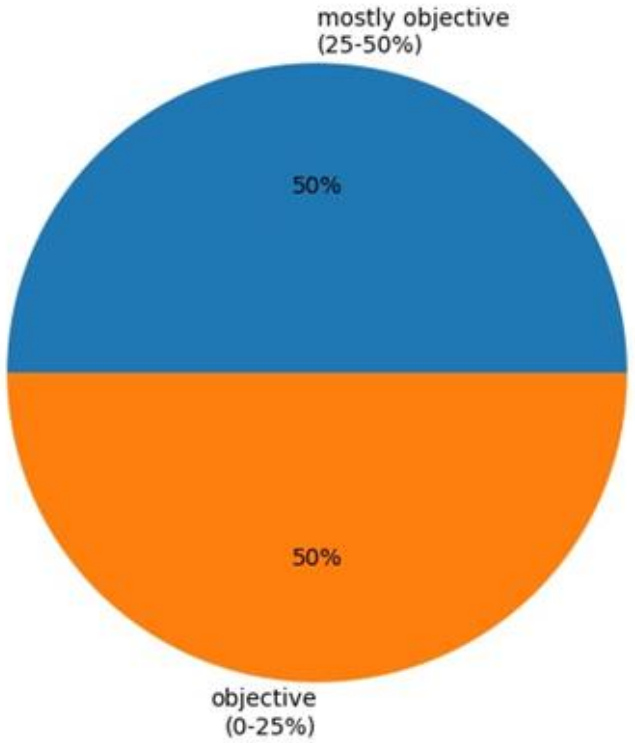
Результати тестування. Твіти про Євробачення





Результати тестування. Статті у телеграм-каналі «Суспільне Кропивницький»





Результати тестування. Статті у «Deutsche Welle»

Веб-сторінка	Тональність	Рівень суб'єктивності
Meinung: Russland für den Wiederaufbau der Ukraine zur Verantwortung ziehen https://www.dw.com/de/meinung-russland-f%C3%BCr-den-wiederaufbau-der-ukraine-zur-verantwortung-ziehen/a-61841071	Нейтральна	32%
EU will Energieimporte aus Russland beenden https://www.dw.com/de/eu-will-energieimporte-aus-russland-beenden/a-61840988	Нейтральна	31%
Was ist ein Gefangenenaustausch? https://www.dw.com/de/was-ist-ein-gefangenenaustausch/a-61830136	Нейтральна	30%
TikTok im Schutzkeller: Die Geschichte von Valeria Shashenok aus Tschernihiw https://www.dw.com/de/valeria-shashenok-tschernihiw-tiktok-lesereise/a-61767290	Нейтральна	34%
Mit regionaler Landwirtschaft gegen die Klimakrise https://www.dw.com/de/mit-regionaler-landwirtschaft-gegen-die-klimakrise/a-61815088	Нейтральна	39%

Відгуки студентів про роботу із SENTIALIZER

- програма не завжди правильно розуміє месседж, але справляється в половині випадків.
- Загалом наші погляди на текст збігалися, протиріч не виникало, що на мою думку, є гарним результатом
- мушу визнати, що незважаючи на непогані результати інструменти не можуть дати 100% правильний висновок і потребують перевірки від користувача.
- SENTIALIZER точно не вписався з емоційністю художнього стилю.
- SENTIALIZER загалом гарно вписався з українською мовою, цікаво, що в графі науковий стиль знаходиться мій власний переклад науко-публ статті і що англ., що укр. SENTIALIZER оцінив практично ідентично
- Сайт зміг розпізнати гумористичне повідомлення

Дякую за увагу!