Лабораторна робота №6

Аналіз настроїв

Мета роботи: Ознайомитись з вирішенням задачі аналізу настроїв та базовими можливостями бібліотеки spaCy.

Короткі теоретичні відомості

Текстові дані, незважаючи на те, що вони дуже неструктуровані, можна розділити на два основних типи документів. Фактичні документи, як правило, зображують певну форму тверджень або фактів без конкретних почуттів чи емоцій. Вони також відомі як об'єктивні документи. З іншого боку, суб'єктивні документи виражають почуття, настрій, емоції та думки.

Ключова ідея полягає в тому, щоб використовувати методи текстової аналітики, обробки природньої мови, машинного навчання та лінгвістики для вилучення важливої інформації або даних із неструктурованого тексту. Це, у свою чергу, може допомогти отримати якісні результати, як-от загальний настрій на позитивній, нейтральній або негативній шкалі, і кількісні результати, як-от полярність настроїв, пропорції суб'єктивності та об'єктивності. Полярність настроїв – це зазвичай числова оцінка, присвоєна позитивним і негативним аспектам текстового документа, яка базується на суб'єктивних параметрах, таких як конкретні слова та фрази, що виражають почуття та емоції.

Моделі аналізу настроїв без учителя використовують добре підібрані бази знань, онтології, лексикони та бази даних, які містять детальну інформацію, що стосується суб'єктивних слів, фраз, включаючи почуття, настрій, полярність, об'єктивність, суб'єктивність тощо. Лексикони містять список слів, пов'язаних із позитивним і негативним настроєм, полярністю (величиною негативного чи позитивного результату), тегами частин мови, класифікаторами суб'єктивності (сильний, слабкий, нейтральний), настроєм, модальністю тощо.

Використаємо лексикон TextBlob та набір даних з відгуками на фільми.

```
for review, sentiment in zip(test_reviews[sample_review_ids], test_sentiments[sample_review_ids]):
  print('REVIEW:', review)
  print('Actual Sentiment:', sentiment)
  print('Predicted Sentiment polarity:', textblob.TextBlob(review).
  sentiment.polarity)
sentiment_polarity = [textblob.TextBlob(review).sentiment.polarity for review in test_reviews]
predicted_sentiments = ['positive' if score >= 0.1 else 'negative' for score in sentiment_polarity]
```

Ще один спосіб побудувати модель для розуміння текстового вмісту та прогнозування настрою текстових відгуків — це використовувати машинне навчання з учителем. Точніше, використовувати класифікаційні моделі для вирішення цієї проблеми.

norm_train_reviews = tn.normalize_corpus(train_reviews)

norm_test_reviews = tn.normalize_corpus(test_reviews)

cv = CountVectorizer(binary=False, min_df=0.0, max_df=1.0, ngram_range=(1,2))

cv_train_features = cv.fit_transform(norm_train_reviews)

tv = TfidfVectorizer(use_idf=True, min_df=0.0, max_df=1.0, ngram_range=(1,2), sublinear_tf=True)

tv_train_features = tv.fit_transform(norm_train_reviews)

cv_test_features = cv.transform(norm_test_reviews)

tv_test_features = tv.transform(norm_test_reviews)

from sklearn.linear_model import SVC, LogisticRegression

lr = LogisticRegression(penalty='l2', max_iter=100, C=1)

svm = SVC(loss='hinge', max_iter=100)

lr.fit(cv_train_features, train_label_names)

predicted=lr.predict(cv_test_features)

Можна вивести матрицю невідповідностей:

from sklearn.metrics import confusion_matrix

confusion_matrix(test_label_names, predicted)

spaCy — бібліотека Python з відкритим кодом для сучасної обробки природньої мови. spaCy постачається з попередньо підготовленими мовними моделями та векторами слів для понад 60 мов.

Коли викликається nlp до тексту, spaCy застосовує деякі етапи обробки. Першим кроком є токенізація для створення об’єкта Doc. Потім об’єкт Doc обробляється далі за допомогою тегера, синтаксичного аналізатора та засобу розпізнавання сутностей.

nlp = spacy.load("en_core_web_sm")

doc = nlp(text)

print ([token.text for token in doc])

Найновіші версії spaCy, після версії 3.4, мають моделі для української мови.

nlp1 = spacy.load("uk_core_news_sm")

Також можна виділити окремі речення:

```
for sent in doc.sents:
   print(sent.text)
```

Для кожного токену можна вивести леми:

```
for token in doc1:
   print(token.text, token.lemma_)
```

За допомогою атрибуту text можна вивести сам текст до обробки:

doc1.text

Окрім того, можна вивести речення

sentences = list(doc.sents)

Сутності:

doc.ents

Також можна перетворити об’єкт doc в json-формат

doc.to_json()
Для об’єкту токен також можна вивести власне токен:
token.text
Його ідентифікатор
token.i
Його індекс
token.idx
Відповідний об’єкт doc
token.doc
Відповідне речення:
token.sent
Та перевірити чи починається це речення з даного токену
token.is_sent_start
Можна перевірити чи є токен стоп-словом:

```
for token in doc:
   print(token, token.is_stop)
```

Завдання до лабораторної роботи

Створити програму, яка:

1. а) Зчитує заданий набір даних, виконує попередню обробку, розбиває дані на навчальні на тестові. Виконує аналіз настроїв за допомогою алгоритмів класифікації (наприклад, логістичної регресії, опорних векторів і т.д.). Виводить матрицю невідповідностей та точність моделі.
   б) Використовує один з готових лексиконів, наприклад Textblob, для аналізу оцінки настроїв. Також розраховує матрицю невідповідностей, та точність моделі.
   в) Обирає три випадкові записи та виводить результати оцінки їх настрою за пунктами а) і б).
2. Виконує завдання відповідно до варіанту засобами бібліотеки spaCy.

Оформити звіт. Звіт повинен містити:

— титульний лист;
— код програми;
— результати виконання коду;

Продемонструвати роботу програми та відповісти на питання стосовно теоретичних відомостей та роботи програми.