Anngladys / AI-Tools-Assignment

Code Issues Pull requests Actions Projects Wiki Security

AI-Tools-Assignment / part2_practical / task3_nlp_spacy.ipynb

Anngladys Winding up 2e937b0 · 2 minutes ago

496 lines (496 loc) · 19.4 KB

Preview Code Blame Raw

In [1]:
```
# Cell 1: Import Libraries
import spacy

# Load the English spaCy model (ensure you've run 'python -m spacy downloa
try:
    nlp = spacy.load("en_core_web_sm")
    print("spaCy model loaded successfully!")
except OSError:
    print("SpaCy model not found. Please run 'python -m spacy download en_
    exit() # Exit if model not loaded
```

```
spaCy model loaded successfully!
```

In [2]:
```
# Cell 2: Define Sample Review Texts
review_texts = [
    "The new iPhone 15 Pro is an amazing device. Apple has outdone themsel
    "This Samsung Galaxy S24 has a terrible battery life. Very disappointe
    "Excellent Bose QuietComfort headphones! Sound quality is superb.",
    "I bought a cheap knockoff charger, it stopped working in a week. Don'
    "The Sony PlayStation 5 is fantastic for gaming, but it's often out of
    "My new Kindle Oasis arrived quickly. It's great for reading, a truly
    "Terrible experience with this Dell XPS laptop, constant crashes."
]

print("Sample review texts defined.")
```

```
Sample review texts defined.
```

In [3]:
```
# Cell 3: Perform Named Entity Recognition (NER)
print("--- Named Entity Recognition (NER) ---")
extracted_entities = []

for i, text in enumerate(review_texts):
    doc = nlp(text)
    entities_in_review = []
    print(f"\nReview {i+1}: \"{text}\"")
    for ent in doc.ents:
        # We're primarily interested in products, organizations, and poten
        if ent.label_ in ["ORG", "PRODUCT", "GPE", "PERSON", "NORP"]: # Ad
            entities_in_review.append({"text": ent.text, "label": ent.labe
            print(f"  - Entity: '{ent.text}' (Type: {ent.label_})")
    extracted_entities.append(entities_in_review)
```

```
--- Named Entity Recognition (NER) ---

Review 1: "The new iPhone 15 Pro is an amazing device. Apple has outdone the
mselves."
  - Entity: 'Apple' (Type: ORG)

Review 2: "This Samsung Galaxy S24 has a terrible battery life. Very disappo
inted with the brand."

Review 3: "Excellent Bose QuietComfort headphones! Sound quality is superb."
  - Entity: 'Bose QuietComfort' (Type: PERSON)

Review 4: "I bought a cheap knockoff charger, it stopped working in a week.
Don't waste your money."
```

```
Don't waste your money."

Review 5: "The Sony PlayStation 5 is fantastic for gaming, but it's often ou
t of stock."
  - Entity: 'Sony' (Type: ORG)
  - Entity: 'PlayStation 5' (Type: PRODUCT)

Review 6: "My new Kindle Oasis arrived quickly. It's great for reading, a tr
uly portable library."
  - Entity: 'Kindle Oasis' (Type: ORG)

Review 7: "Terrible experience with this Dell XPS laptop, constant crashes."
  - Entity: 'Dell XPS' (Type: ORG)
```

In [4]:

```python
# Cell 4: Analyze Sentiment (Rule-Based Approach)
print("\n--- Sentiment Analysis (Rule-Based) ---")

positive_words = ["amazing", "excellent", "superb", "fantastic", "great",
negative_words = ["terrible", "disappointed", "stopped working", "waste",

def analyze_sentiment_rule_based(text):
    text_lower = text.lower()
    pos_score = sum(1 for word in positive_words if word in text_lower)
    neg_score = sum(1 for word in negative_words if word in text_lower)

    if pos_score > neg_score:
        return "Positive"
    elif neg_score > pos_score:
        return "Negative"
    else:
        return "Neutral" # Or if pos_score == neg_score

for i, text in enumerate(review_texts):
    sentiment = analyze_sentiment_rule_based(text)
    print(f"\nReview {i+1}: \"{text}\"")
    print(f"  - Sentiment: {sentiment}")
```

```
--- Sentiment Analysis (Rule-Based) ---

Review 1: "The new iPhone 15 Pro is an amazing device. Apple has outdone the
mselves."
  - Sentiment: Positive

Review 2: "This Samsung Galaxy S24 has a terrible battery life. Very disappo
inted with the brand."
  - Sentiment: Negative

Review 3: "Excellent Bose QuietComfort headphones! Sound quality is superb."
  - Sentiment: Positive

Review 4: "I bought a cheap knockoff charger, it stopped working in a week.
Don't waste your money."
  - Sentiment: Negative

Review 5: "The Sony PlayStation 5 is fantastic for gaming, but it's often ou
t of stock."
  - Sentiment: Positive

Review 6: "My new Kindle Oasis arrived quickly. It's great for reading, a tr
uly portable library."
  - Sentiment: Positive

Review 7: "Terrible experience with this Dell XPS laptop, constant crashes."
```

```
- Sentiment: Negative
```

In [5]:

```python
# Cell 1: Import Libraries
import spacy
import pandas as pd
import random

print(f"spaCy Version: {spacy.__version__}")
print("Libraries imported successfully!")
```

```
spaCy Version: 3.8.7
Libraries imported successfully!
```

In [6]:

```python
# Cell 2: Load spaCy English Model
try:
    # Load the small English model
    nlp = spacy.load("en_core_web_sm")
    print("spaCy 'en_core_web_sm' model loaded successfully.")
except OSError:
    print("spaCy model 'en_core_web_sm' not found. Downloading...")
    spacy.cli.download("en_core_web_sm")
    nlp = spacy.load("en_core_web_sm")
    print("spaCy model 'en_core_web_sm' downloaded and loaded successfully
```

```
spaCy 'en_core_web_sm' model loaded successfully.
```

In [7]:

```python
# Cell 3: Sample Text Data (Amazon Reviews style)
amazon_reviews = [
    "The product is excellent! Very happy with the purchase.",
    "Battery life is terrible, died after 2 hours. Very disappointed.",
    "Works as expected, good value for money. Highly recommended.",
    "This is the worst item I've ever bought. A complete waste of money.",
    "It's okay, not great, not bad. Just mediocre.",
    "Fantastic performance, totally exceeded my expectations!",
    "Wish it had more features, but it's decent for the price.",
    "The delivery was fast, but the item was damaged.",
    "Absolutely love this! The design is sleek and it's so easy to use.",
    "Received a broken one. Customer service was unhelpful."
]

print("Sample Amazon reviews loaded.")
```

```
Sample Amazon reviews loaded.
```

In [8]:

```python
# Cell 4: Tokenization, POS Tagging, and Lemmatization
print("--- Tokenization, POS Tagging, and Lemmatization ---")
for i, text in enumerate(amazon_reviews[:3]): # Process first 3 reviews fo
    doc = nlp(text)
    print(f"\nReview {i+1}: '{text}'")
    print(f"{'Token':<15} {'Lemma':<15} {'POS':<10} {'Is Alpha?':<10} {'St
    print("-" * 70)
    for token in doc:
        print(f"{str(token):<15} {token.lemma_:<15} {token.pos_:<10} {str(
```

```
--- Tokenization, POS Tagging, and Lemmatization ---

Review 1: 'The product is excellent! Very happy with the purchase.'
```

```
Token           Lemma           POS         Is Alpha?   Stopword?
----------------------------------------------------------------------
The             the             DET         True        True
product         product         NOUN        True        False
is              be              AUX         True        True
excellent       excellent       ADJ         True        False
!               !               PUNCT       False       False
Very            very            ADV         True        True
happy           happy           ADJ         True        False
with            with            ADP         True        True
the             the             DET         True        True
purchase        purchase        NOUN        True        False
.               .               PUNCT       False       False

Review 2: 'Battery life is terrible, died after 2 hours. Very disappointed.'
Token           Lemma           POS         Is Alpha?   Stopword?
----------------------------------------------------------------------
Battery         battery         NOUN        True        False
life            life            NOUN        True        False
is              be              AUX         True        True
terrible        terrible        ADJ         True        False
,               ,               PUNCT       False       False
died            die             VERB        True        False
after           after           ADP         True        True
2               2               NUM         False       False
hours           hour            NOUN        True        False
.               .               PUNCT       False       False
Very            very            ADV         True        True
disappointed    disappointed    ADJ         True        False
.               .               PUNCT       False       False

Review 3: 'Works as expected, good value for money. Highly recommended.'
Token           Lemma           POS         Is Alpha?   Stopword?
----------------------------------------------------------------------
Works           work            NOUN        True        False
as              as              SCONJ       True        True
expected        expect          VERB        True        False
,               ,               PUNCT       False       False
good            good            ADJ         True        False
value           value           NOUN        True        False
for             for             ADP         True        True
money           money           NOUN        True        False
.               .               PUNCT       False       False
Highly          highly          ADV         True        False
recommended     recommend       VERB        True        False
.               .               PUNCT       False       False
```

In [9]:

```python
# Cell 5: Named Entity Recognition (NER)
print("\n--- Named Entity Recognition (NER) ---")
for i, text in enumerate(amazon_reviews):
    doc = nlp(text)
    if doc.ents:
        print(f"\nReview {i+1}: '{text}'")
        for ent in doc.ents:
            print(f"  Entity: {ent.text}, Type: {ent.label_}, SpaCy Explan
    else:
        print(f"\nReview {i+1}: '{text}' - No entities found.")
```

```
--- Named Entity Recognition (NER) ---

Review 1: 'The product is excellent! Very happy with the purchase.' - No ent
ities found.
```

```
Review 2: 'Battery life is terrible, died after 2 hours. Very disappointed.'
  Entity: 2 hours, Type: TIME, SpaCy Explanation: Times smaller than a day

Review 3: 'Works as expected, good value for money. Highly recommended.' - N
o entities found.

Review 4: 'This is the worst item I've ever bought. A complete waste of mone
y.' - No entities found.

Review 5: 'It's okay, not great, not bad. Just mediocre.' - No entities foun
d.

Review 6: 'Fantastic performance, totally exceeded my expectations!'
  Entity: Fantastic, Type: NORP, SpaCy Explanation: Nationalities or religio
us or political groups

Review 7: 'Wish it had more features, but it's decent for the price.' - No e
ntities found.

Review 8: 'The delivery was fast, but the item was damaged.' - No entities f
ound.

Review 9: 'Absolutely love this! The design is sleek and it's so easy to us
e.' - No entities found.

Review 10: 'Received a broken one. Customer service was unhelpful.' - No ent
ities found.
```

In [10]:

```python
# Cell 6: Basic Rule-Based Sentiment Analysis (Illustrative - very simple)
print("\n--- Basic Rule-Based Sentiment Analysis (Illustrative) ---")

positive_words = ["excellent", "happy", "good", "recommended", "fantastic"
negative_words = ["terrible", "disappointed", "worst", "waste", "mediocre"

def simple_sentiment(text):
    doc = nlp(text.lower()) # Process lowercase text
    sentiment_score = 0
    for token in doc:
        if token.text in positive_words:
            sentiment_score += 1
        elif token.text in negative_words:
            sentiment_score -= 1
    if sentiment_score > 0:
        return "Positive"
    elif sentiment_score < 0:
        return "Negative"
    else:
        return "Neutral"

for i, review in enumerate(amazon_reviews):
    sentiment = simple_sentiment(review)
    print(f"Review {i+1}: '{review}'\n  Sentiment: {sentiment}\n")

print("\nNote: This is a very simplistic rule-based sentiment analysis.")
print("It lacks context understanding, sarcasm detection, and nuances. For
print("Review: 'This is great, another broken item!' (Should be Negative)"
doc_sarcasm = nlp("This is great, another broken item!")
print(f"  Simple rule-based analysis: {simple_sentiment(str(doc_sarcasm))}
print("\nAdvanced NLP (like machine learning models or deep learning) is n
```

```
--- Basic Rule-Based Sentiment Analysis (Illustrative) ---
```

```
Review 1: 'The product is excellent! Very happy with the purchase.'
  Sentiment: Positive

Review 2: 'Battery life is terrible, died after 2 hours. Very disappointed.'
  Sentiment: Negative

Review 3: 'Works as expected, good value for money. Highly recommended.'
  Sentiment: Positive

Review 4: 'This is the worst item I've ever bought. A complete waste of mone
y.'
  Sentiment: Negative
```