Oppgave 4, Språkmodeller og ordklasser (løsningsforslag)

January 13, 2022

1 NLTK og språkmodeller

1. Setningsmarkørene merkes som `None` i stedet for `<s>` og `<\s>`. Det er to stk. fordi vi vi bruker trigram.

2. For å beregne sannsynligheten multipliserer vi sammen sannsynligheten for hvert trigram. Dette er mulig fordi vi antar at hvert trekk er uavhengig. NumPy-metoden `np.prod()` tar inn en liste med tall som argument og multipliserer sammen hvert tall.

2 Teori om språkmodeller

$P(Katherine| < s >) \cdot P(spiller|Katherine) \cdot P(pinao|spiller) \cdot P(< \backslash s > |piano)$

$\frac{Count(<s>,Katherine)}{Count(<s>)} \cdot \frac{Count(Katherine,spiller)}{Count(Katherine)} \cdot \frac{Count(spiller,pinao)}{Count(spiller)} \cdot \frac{Count(pinao,<\backslash s>)}{Count(pinao)}$

$\frac{1}{3} \cdot \frac{1}{1} \cdot \frac{2}{3} \cdot \frac{3}{3} = \frac{6}{27} = \frac{2}{9}$

3 Flertydighet

1. Vi har flere typer flertydighet. **Leksikalsk (semantisk) flertydighet** vil si at et leksem har flere betydninger. Dette inkluderer homonymi, at ulike leksemer har samme form eller lyd, og polysemi, at ulike leksemer har forskjellige betydninger. Eksempel på norsk: *tre (verb)*, *tre (substantiv)*, *tre (determinativ, kvantor)*. Eksempel på engelsk: *see*, *sea*. I tillegg har vi **strukturell (syntaktisk) flertydighet**, som vil si at en frase eller setning kan tolkes på flere måter med ulik betydning. Eksempel: *Pensjonist [jaget [inntrenger med gevær]]* kontra *Pensjonist [jaget [inntrenger] med gevær]*.

2.

[1]:
```
from nltk.corpus import brown

brown_news = brown.tagged_words(categories='news')

# Skriver ut 10 ord for å sjekke
print(brown_news[:10])
```

```
[('The', 'AT'), ('Fulton', 'NP-TL'), ('County', 'NN-TL'), ('Grand', 'JJ-TL'),
('Jury', 'NN-TL'), ('said', 'VBD'), ('Friday', 'NR'), ('an', 'AT'),
('investigation', 'NN'), ('of', 'IN')]
```

3.

[2]:
```
from collections import defaultdict

# Oppretter en defaultdict med standardverdi tom liste
tags = defaultdict(lambda: [])

# Itererer gjennom hvert ord og tag samtidig
# Dette er mulig fordi de er lagret tuppelvis
for word, tag in brown_news:

    # Håndterer store/små bokstaver slik at f.eks.
    # "The" og "the" ikke regnes som ulike typer
    word = word.lower()

    # Sjekker først om vi allerede har lagret tag-en for ordet
    # For nye ord vil en ny nøkkel opprettes med standardverdien
    if tag not in tags[word]:

        # Hvis ikke: legg til
        tags[word].append(tag)

# Oppretter en teller
count = 0

# Itererer over tag-ene
for list_of_tags in tags.values():

    #  Hvis antall tag-er er mer enn 1 ...
    if len(list_of_tags) > 1:

        # ... øker vi telleren
        count += 1

print('Antall flertydige typer:', count)
```

```
Antall flertydige typer: 2166
```

4.

[3]:
```
tags_as_list = list(tags.items())

# Husk at sort() modifiserer listen
# direkte uten å returnere noe
tags_as_list.sort(key=lambda x: len(x[1]), reverse=True)

print(tags_as_list[:5])
```

```
[('to', ['TO', 'IN', 'IN-HL', 'TO-HL', 'IN-TL', 'TO-TL', 'NPS']), ('house',
```

```
['NN-TL', 'NN', 'NN-HL', 'NP-TL-HL', 'NP', 'NN-TL-HL', 'VB']), ('home', ['NR',
'NN', 'NN-HL', 'NN-TL', 'NR-HL', 'NP']), ('that', ['CS', 'WPS', 'DT', 'QL',
'WPO']), ('for', ['IN', 'IN-TL', 'IN-HL', 'CS', 'RB'])]
```

5.

[4]:
```
unique_tags = {word_and_tag[1] for word_and_tag in brown_news}

print('Antall unike tag-er:', len(unique_tags))
```

```
Antall unike tag-er: 218
```

6.

[5]:
```
def freqs(target_word):

    target_word = target_word.lower()

    # Oppretter en defaultdict
    tag_counts = defaultdict(lambda: 0)

    # Itererer gjennom ord og tag-er i korpus
    for word, tag in brown_news:

        # Sjekker ordet
        if word == target_word:

            # Teller tag-en
            tag_counts[tag] += 1

    # Skriver ut oversiktlig
    print(target_word)
    for tag, count in tag_counts.items():
        print(f'{tag}:\t{count}')
    print('\n')

# Test
freqs('run')
```

```
run
VB:     11
NN:     20
VBN:    4
```

7.

[6]:
```
freqs('to')
freqs('house')
```

```
freqs('home')
```

```
to
TO:     1222
IN:     880
IN-HL:  5
TO-HL:  6
IN-TL:  2
NPS:    1


house
NN:     24
NN-HL:  1
VB:     1


home
NR:     53
NN:     72
NN-HL:  1
NR-HL:  1
```

Vi observerer at distribusjonen er veldig ujevn.