IN2110 - Oblig2b

Stian Carlsen Swärd (stiancsw)

May 10, 2020

Del 1: Maskinoversettelse

a) Utvikling av en frasetabell

Oppgaver:

1. Hva er modellens oversettelse for linjene 10-15?

```
because these rings bargen the force and the will to lead every people .
but they all cheated .
because a ring was made .
in mordor , in the flash-over of the schicksalsberges forging of the dark lord sauro
in this ring , his cruelty , his cattiness and his will , went to burdensome all lif
a ring to enslaving them .
```

2. Finn et eksempel hvor bruk av ordboken førte til en dårlig oversettelse.
   Oversettelsen for linje 3 skulle vært `''i feel it in the earth .''` men
   er i stedet `''i feel it in the world .''`

3. Lag en liste med minst 10 ord som systematisk er feil oversatt.

   1) `beutlin|||baggins`
   2) `elben|||elves`
   3) `langerwartete|||long awaited`
   4) `namenloses|||nameless`
   5) `nebelgebirge|||misty mountains`
   6) `reinsten|||fairest`
   7) `schicksalsberges|||mount doom`
   8) `unwahrscheinlichsten|||most unlikely`
   9) `vergiftete|||poisoned`
   10) `weisesten|||wisest`

b) Evaluering

Oppgaver:

1. Fyll ut `compute_precision(ref_file, output_file, ngram_order)`:

```
ref_sentences = get_sentences(reference_file)
output_sentences = get_sentences(output_file)

total_ngrams = 0
matching_ngrams = 0

for ref, mtl in zip(ref_sentences, output_sentences):
    ref_ngrams = [ref[n:n+ngram_order] for n in range(len(ref) - ngram_order + 1)]
    for n in range(len(mtl) - ngram_order + 1):
        if mtl[n:n+ngram_order] in ref_ngrams:
            matching_ngrams += 1
        total_ngrams += 1

return matching_ngrams / total_ngrams
```

2. Fyll ut `compute_brevity_penalty(ref_file, output_file)`:

```
ref_sentences = get_sentences(reference_file)
output_sentences = get_sentences(output_file)

ref_words = 0
mtl_words = 0

for ref, mtl in zip(ref_sentences, output_sentences):
    ref_words += len(ref)
    mtl_words += len(mtl)

return min(1, mtl_words / ref_words)
```

3. og 4.:
   BLEU-score for maskinoversettelse av `lotr.de` med og uten frasetabellen `de-en.txt`:

Frasetabell?	BLEU-score
NEI	0.232
JA	0.236

Del 2: Interaktive systemer

Oppgaver

1. Fyll ut `get_tf_idf(self, utterance)`:

```
vector = {}
for word in set(utterance):
    tf = utterance.count(word) / len(utterance)
    df = 0
    idf = np.log(len(self.utterances) / self.doc_freqs[word])
    vector[word] = tf * idf

return vector
```

2. Fyll ut `compute_cosine(self, tf_idf1, tf_idf2)`:

```
# Create a 'corpus' of all the words in tf_idf1 and tf_idf2
common_words = set(tf_idf1.keys())
common_words.update(set(tf_idf2.keys()))

# Create numpy arrays of shape (1, len(common_words))
# and fill with values
tf_idf1_vec = np.array([tf_idf1.get(word, 0) for word in common_words])
tf_idf2_vec = np.array([tf_idf2.get(word, 0) for word in common_words])

# Standard cosine similarity using numpy dot product
return tf_idf1_vec @ tf_idf2_vec / (self._get_norm(tf_idf1) * self._get_norm(tf_idf2
```

3. Fyll ut `get_response(self, query)`:

```
# If the query is a string, we first tokenise it
if type(query)==str:
    query = self._tokenise(query)

# Convert query to tf_idf vector
query = self.get_tf_idf(query)

# Compare query vector to corpus, keeping best match
best_cosine_similarity = 0
best_index = 0
for n in range(len(self.tf_idfs) - 1):
    cosine_similarity = self.compute_cosine(query, self.tf_idfs[n])
    if cosine_similarity > best_cosine_similarity:
        best_cosine_similarity = cosine_similarity
        best_index = n

return ' '.join(self.utterances[best_index + 1])
```