# stat29000project04solutions

February 5, 2020

## 1 STAT29000 Project 4 Solutions

### 1.1 Question 1

[9]:
```
from typing import Tuple

from media.rottentomatoes.utilities import search
from media.rottentomatoes.reviews import get_reviews
from media.rottentomatoes import datasets as ds
import string

test_corpus, _ = get_reviews("the_girl_with_the_dragon_tattoo", 50)
test_document = test_corpus[0]
test_terms = ["a", "the", "years", "absolute"]

def tf(document: str, terms: Tuple[str, ...]):
    document = document.lower()
    return [document.translate(document.maketrans('', '', string.punctuation)).
 ↪split().count(term.lower()) for term in terms]
tf(test_document, test_terms)

my_terms = ('the', 'a', 'farm',)
my_document = 'I went to the farm and a boar charged at me. I will not return␣
 ↪to the farm.'
tf(my_document, my_terms)
```

[9]:
```
[2, 1, 2]
```

[11]:
```
from math import log
from typing import Tuple

test_corpus, _ = get_reviews("the_girl_with_the_dragon_tattoo", 50) # 241␣
 ↪reviews

def idf(corpus: Tuple[str, ...], terms: Tuple[str, ...]):

    def _dt(corpus: Tuple[str, ...], term: str):
```

```
        """
        Helper function that returns the number of documents
        in the provided corpus where the term appears.
        """
        return sum(tuple(document.lower().translate(document.lower().
↪maketrans('', '', string.punctuation)).split().count(term.lower()) > 0 for␣
↪document in corpus))

    return [log(len(corpus)/_dt(corpus, term)) for term in terms]

corpus = (
  "This is a sentence in the corpus.",
  "Each of these is a document in the corpus.",
  "Another sentence is here.",
  "The last sentence made no sense.",
  "Neither did that last one, another confusing one.",
  "Last one for sure.",
)

terms = ("sentence", "is", "a", "the", "that", "one", "last")

print(idf(corpus, terms))

# as of right now, this should create a ZeroDivisionError since
# "never" never occurs anywhere in any document in the corpus
terms = ("sentence", "is", "a", "the", "that", "one", "last", "never")
idf(corpus, terms)
```

```
[0.6931471805599453, 0.6931471805599453, 1.0986122886681098, 0.6931471805599453,
1.791759469228055, 1.0986122886681098, 0.6931471805599453]
```

```
      ␣
↪---------------------------------------------------------------------------

       ZeroDivisionError                         Traceback (most recent call␣
↪last)

       <ipython-input-11-ef2ffbd98e9e> in <module>
        31 # "never" never occurs anywhere in any document in the corpus
        32 terms = ("sentence", "is", "a", "the", "that", "one", "last",␣
↪"never")
   ---> 33 idf(corpus, terms)


       <ipython-input-11-ef2ffbd98e9e> in idf(corpus, terms)
```

```
        13         return sum(tuple(document.lower().translate(document.lower().
↪maketrans('', '', string.punctuation)).split().count(term.lower()) > 0 for␣
↪document in corpus))
        14
   ---> 15     return [log(len(corpus)/_dt(corpus, term)) for term in terms]
        16
        17 corpus = (


       <ipython-input-11-ef2ffbd98e9e> in <listcomp>(.0)
        13         return sum(tuple(document.lower().translate(document.lower().
↪maketrans('', '', string.punctuation)).split().count(term.lower()) > 0 for␣
↪document in corpus))
        14
   ---> 15     return [log(len(corpus)/_dt(corpus, term)) for term in terms]
        16
        17 corpus = (


       ZeroDivisionError: division by zero
```

[13]:
```
def idf(corpus: Tuple[str, ...], terms: Tuple[str, ...], smooth: bool = True):

    def _dt(corpus: Tuple[str, ...], term: str):
        """
        Helper function that returns the number of documents
        in the provided corpus where the term appears.
        """
        return sum(tuple(document.lower().translate(document.lower().
 ↪maketrans('', '', string.punctuation)).split().count(term.lower()) > 0 for␣
 ↪document in corpus))

    if smooth:
        return [log((1+len(corpus))/(1+_dt(corpus, term))) for term in terms]

    return [log(len(corpus)/_dt(corpus, term)) for term in terms]

idf(test_corpus, test_terms)

corpus = (
  "This is a sentence in the corpus.",
  "Each of these is a document in the corpus.",
  "Another sentence is here.",
  "The last sentence made no sense.",
  "Neither did that last one, another confusing one.",
  "Last one for sure.",
```

```
)

terms = ("sentence", "is", "a", "the", "that", "one", "last")

print(idf(corpus, terms))

# this should now work
terms = ("sentence", "is", "a", "the", "that", "one", "last", "never")
idf(corpus, terms)
```

```
[0.5596157879354227, 0.5596157879354227, 0.8472978603872037, 0.5596157879354227,
1.252762968495368, 0.8472978603872037, 0.5596157879354227]
```

[13]:
```
[0.5596157879354227,
 0.5596157879354227,
 0.8472978603872037,
 0.5596157879354227,
 1.252762968495368,
 0.8472978603872037,
 0.5596157879354227,
 1.9459101490553132]
```

### 1.2 Question 2

[14]:
```
def tfidf(corpus: Tuple[str, ...], terms: Tuple[str, ...]):
    result = []
    term_idfs = idf(corpus, terms)
    for idx, document in enumerate(corpus):
        result.append([tf*idf for tf, idf in zip(tf(document, terms),␣
 ↪term_idfs)])

    return result

tfidf(test_corpus, test_terms)

corpus = [
  "This is a sentence in the corpus.",
  "Another sentence is here.",
  "The last sentence made no sense.",
  "Neither did that last one, another confusing one.",
  "Last one for sure.",
]

terms = ["sentence", "is", "a", "the", "that", "one", "last"]

tfidf(corpus, terms)
```

[14]:
```
[[0.4054651081081644,
  0.6931471805599453,
  1.0986122886681098,
  0.6931471805599453,
  0.0,
  0.0,
  0.0],
 [0.4054651081081644, 0.6931471805599453, 0.0, 0.0, 0.0, 0.0, 0.0],
 [0.4054651081081644,
  0.0,
  0.0,
  0.6931471805599453,
  0.0,
  0.0,
  0.4054651081081644],
 [0.0,
  0.0,
  0.0,
  0.0,
  1.0986122886681098,
  1.3862943611198906,
  0.4054651081081644],
 [0.0, 0.0, 0.0, 0.0, 0.0, 0.6931471805599453, 0.4054651081081644]]
```

[15]:
```
import string
from stop_words import get_stop_words

def parse_doc(document: str):
    document = document.lower()
    stop_words = get_stop_words('english')
    return [word for word in document.translate(document.maketrans('', '',␣
 ↪string.punctuation)).split() if word not in stop_words]

parse_doc("This is a test. Okay, we're set.")
```

[15]:
```
['test', 'okay', 'set']
```

[16]:
```
def corpus_terms(corpus: Tuple[str, ...]):
    return list(set(parse_doc(' '.join(tuple(document for document in␣
 ↪corpus)))))


corpus = [
  "This is a sentence in the corpus.",
  "Another sentence is here.",
  "The last sentence made no sense.",
  "Neither did that last one, another confusing one.",
```

```
  "Last one for sure.",
]
corpus_terms(corpus)
```

[16]:
```
['another',
 'sentence',
 'neither',
 'corpus',
 'confusing',
 'one',
 'made',
 'sure',
 'sense',
 'last']
```

[17]:
```
def idf(corpus: Tuple[str, ...], terms: Tuple[str, ...], smooth: bool = True):

    def _dt(corpus: Tuple[str, ...], term: str):
        """
        Helper function that returns the number of documents
        in the provided corpus where the term appears.
        """
        return sum(tuple(term.lower() in parse_doc(document) for document in␣
 ↪corpus))
    if smooth:
        return [log((1+len(corpus))/(1+_dt(corpus, term))) for term in terms]

    return [log(len(corpus)/_dt(corpus, term)) for term in terms]

def tf(document: str, terms: Tuple[str, ...]):
    return [parse_doc(document).count(term.lower()) for term in terms]
```