



# Query Likelihood Model

# Content

- ❖ Query Likelihood Model concept
- ❖ The Aim of this project
- ❖ Overview of the techniques used in the project
- ❖ Code
- ❖ Output
- ❖ Conclusion

# Query Likelihood Model

- ❖ It is one of the language models.
- ❖ The goal of a language model is to assign a probability to a sequence of words by means of a probability distribution
- ❖ The query likelihood model is a language model used in information retrieval.
- ❖ A language model is constructed for each document in the collection. It is then possible to rank each document by the probability of specific documents given a query.

## The Aim of this project

The aim of this project is to build retrieval systems using the query likelihood model, which will show the percentage of the probability of the query being in each file.

# Overview of the techniques used in the project

- ❖ Query Likelihood:  $p(D|Q) \stackrel{rank}{=} P(Q|D)P(D)$
- ❖ Removes all punctuations from a string.
- ❖ Stopping word
- ❖ Tokenization
- ❖ Stemming/Lemmatization



Implementation

# Code

First, we need to initialize and import all the libraries we need in the code.

```
import nltk
nltk.download('wordnet')
import re
import numpy as np
from nltk.stem import WordNetLemmatizer
from nltk.corpus import stopwords
nltk.download('stopwords')

nltk.download('punkt')
```

# Code

Then we start by defining some of the functions we want to use on the documents.

```
sno = nltk.stem.SnowballStemmer('english')
```

```
lemmatizer = WordNetLemmatizer()
```

```
def remove_punc(string):  
    →punctuations = '''-~!@#$%^&*~?.,/[]{};:'"\<>./?@#$$%^&*~?.[?]''  
  
    →for x in string:  
    →    →if x in punctuations:  
    →    →    →string = string.replace(x, " ")  
  
    →return re.sub(' +', ' ', string.lower().strip())
```



# Code

At this point we will start reading the files.

```
with open("corpus.txt", 'r') as f1:
    docs = f1.readlines()
with open("query.txt", 'r') as f2:
    queries = f2.readlines()

doc_0 = remove_punc(docs[0])
print(doc_0)
word_list = nltk.word_tokenize(doc_0)
doc_0 = ' '.join([lemmatizer.lemmatize(w) for w in word_list])
print(doc_0)
```

# Code

```
corpus = []
query = []
for doc in docs:
    tmp = remove_punc(doc)
    word_list = nltk.word_tokenize(tmp)
    lemmas = [lemmatizer.lemmatize(w) for w in word_list] # lemmatization
    stemmed = " ".join([sno.stem(w) for w in lemmas if w not in stopwords.words('english')]) # stemming of lemmas
    corpus.append(stemmed)

for q in queries:
    tmp = remove_punc(q)
    word_list = nltk.word_tokenize(tmp)

    lemmas = [lemmatizer.lemmatize(w) for w in word_list] # lemmatization
    stemmed = " ".join([sno.stem(w) for w in lemmas if w not in stopwords.words('english')])
    query.append(stemmed)
```

# Unigram Model

Now we start executing Query Likelihood Model.

## Unigram model

```
for q in query:
    q_words = nltk.word_tokenize(q)
    prob = []
    for doc in corpus:
        doc_words = nltk.word_tokenize(doc)
        d = []
        p = 1
        for word in q_words:
            p *= doc_words.count(word) / len(doc_words) + 1e-7
        prob.append(p)
    max_prob_idx = np.argmax(prob)
    print(f'query is: {q}')
    print(f'relevant doc: {corpus[max_prob_idx]}')
    print(prob)
```

# Bigram model

## Bigram model

```
corpus = []
query = []
for doc in docs:
    tmp = remove_punc(doc)
    word_list = nltk.word_tokenize(tmp)
    lemmas = [lemmatizer.lemmatize(w) for w in word_list] # lemmatization
    stemmed = [sno.stem(w) for w in lemmas if w not in stopwords.words('english')]
    corpus.append(stemmed)

for q in queries:
    tmp = remove_punc(q)
    word_list = nltk.word_tokenize(tmp)

    lemmas = [lemmatizer.lemmatize(w) for w in word_list] # lemmatization
    stemmed = [sno.stem(w) for w in lemmas if w not in stopwords.words('english')]
    query.append(stemmed)

def create_bigrams(words):
    """
    Function that creates a list of bigrams from a list of words in a document.
    Returns a list.
    """
    bigrams = []

    for word1, word2 in zip(words, words[1:]):
        bigrams.append(" ".join([word1, word2]))
    return bigrams
```

# Bigram model

```
for q in query:
    query_bigrams = create_bigrams(q)

    prob = []
    for doc in corpus:
        doc_bigrams = create_bigrams(doc)
        p = 1
        for bigram in query_bigrams:
            p *= doc_bigrams.count(bigram) / len(doc_bigrams) + 1e-7

        prob.append(p)
    max_prob_idx = np.argmax(prob)

    q = " ".join(q)
    d = " ".join(corpus[max_prob_idx])
    print(f'query is: {q}')
    print(f'relevant doc: {d}')
    print(prob)
    print()
```

# The Query

- python is an iterative language
- Force
- Thermodynamics
- World War 2 is a historic event
- nuclear bomb

# Output

## The outputs for Unigram

query is: python iter languag

relevant doc: python interpret high level general purpos program languag design philosophi emphas code readabl use signific indent l  
anguag construct well object orient approach aim help program write clear logic code small larg scale project python dynam type gar  
bag collect support multipl program paradigm includ structur particular procedur object orient function program often describ batter  
i includ languag due comprehens standard librari guid van rossum began work python late 1980s successor abc program languag first r  
eleas 1991 python 0 9 0 python 2 0 wa releas 2000 introduc new featur list comprehens cycl detect garbag collect system addit refer  
count python 3 0 wa releas 2008 wa major revis languag complet backward compat python 2 wa discontinu version 2 7 18 2020 python con  
sist rank one popular program languag

[9.9999999999999997e-22, 7.407507407407407e-17, 3.0234427059812774e-10, 8.849657522123892e-17, 9.9999999999999997e-22, 9.9999999999999997e-22]

query is: forc

relevant doc: physic forc influenc chang motion object forc caus object mass chang veloc e g move state rest e acceler forc also des  
crib intuit push pull forc ha magnitud direct make vector quantiti measur si unit newton n forc repres symbol f former p origin form  
newton second law state net forc act upon object equal rate momentum chang time mass object constant law impli acceler object direct  
proport net forc act object direct net forc invers proport mass object concept relat forc includ thrust increas veloc object drag de  
creas veloc object torqu produc chang rotat speed object extend bodi part usual appli forc adjac part distribut forc bodi intern mec  
han stress intern mechan stress caus acceler bodi forc balanc one anoth pressur distribut mani small forc appli area bodi simpl type  
stress unbalanc caus bodi acceler stress usual caus deform solid materi flow fluid

[0.09027787777777778, 1e-07, 1e-07, 1e-07, 1e-07, 0.006802821088435374]

query is: thermodynam

relevant doc: thermodynam heat energi transfer thermodynam system mechan thermodynam work transfer matter various mechan energi tran  
sfer defin heat state next section articl like thermodynam work heat transfer process involv one system properti one system thermody  
nam energi transfer heat contribut chang system cardin energi variabl state exampl intern energi exampl enthalpi distinguish ordinari  
i languag concept heat properti isol system quantiti energi transfer heat process amount transfer energi exclud thermodynam work wa  
done energi contain matter transfer precis definit heat necessari occur path doe includ transfer matter though immedi definit specia  
l kind process quantiti energi transfer heat measur effect state interact bodi exampl respect special circumst heat transfer measur  
amount ice melt chang temperatur bodi surround system method call calorimetri convent symbol use repres amount heat transfer thermod  
ynam process q amount energi transfer si unit heat joul j

[1e-07, 0.05185195185185185, 1e-07, 1e-07, 1e-07, 1e-07]

# Output

query is: world war 2 histor event

relevant doc: nuclear weapon also known atom bomb atom bomb nuclear bomb nuclear warhead colloqui bomb nuke explos devic deriv destr uct forc nuclear reaction either fission fission bomb combin fission fusion reaction thermonuclear bomb bomb type releas larg quanti ti energi relat small amount matter first test fission atom bomb releas amount energi approxim equal 20 000 ton tnt 84 tj 1 first th ermonuclear hydrogen bomb test releas energi approxim equal 10 million ton tnt 42 pj nuclear bomb yield 10 ton tnt w54 50 megaton ts ar bomba see tnt equival thermonuclear weapon weigh littl 2 400 pound 1 100 kg releas energi equal 1 2 million ton tnt 5 0 pj 2 nucl ear devic larger convent bomb devast entir citi blast fire radiat sinc weapon mass destruct prolifer nuclear weapon focus intern rel at polici nuclear weapon deploy twice war unit state japanes citi hiroshima nagasaki 1945 world war ii

[9.9999999999999996e-36, 9.9999999999999996e-36, 2.3809623809523803e-30, 9.9999999999999996e-36, 2.5060800062751736e-24, 1.888908627860255e-20]

query is: nuclear bomb

relevant doc: nuclear weapon also known atom bomb atom bomb nuclear bomb nuclear warhead colloqui bomb nuke explos devic deriv destr uct forc nuclear reaction either fission fission bomb combin fission fusion reaction thermonuclear bomb bomb type releas larg quanti ti energi relat small amount matter first test fission atom bomb releas amount energi approxim equal 20 000 ton tnt 84 tj 1 first th ermonuclear hydrogen bomb test releas energi approxim equal 10 million ton tnt 42 pj nuclear bomb yield 10 ton tnt w54 50 megaton ts ar bomba see tnt equival thermonuclear weapon weigh littl 2 400 pound 1 100 kg releas energi equal 1 2 million ton tnt 5 0 pj 2 nucl ear devic larger convent bomb devast entir citi blast fire radiat sinc weapon mass destruct prolifer nuclear weapon focus intern rel at polici nuclear weapon deploy twice war unit state japanes citi hiroshima nagasaki 1945 world war ii

[9.9999999999999998e-15, 9.9999999999999998e-15, 9.9999999999999998e-15, 9.9999999999999998e-15, 7.8316438258884e-05, 0.004072390175399884]



# Output

## The outputs for Bigram

```
query is: python iter languag
relevant doc: physic forc influenc chang motion object forc caus object mass chang veloc e g move state rest e acceler forc also des
crib intuit push pull forc ha magnitud direct make vector quantiti measur si unit newton n forc repres symbol f former p origin form
newton second law state net forc act upon object equal rate momentum chang time mass object constant law impli acceler object direct
proport net forc act object direct net forc invers proport mass object concept relat forc includ thrust increas veloc object drag de
creas veloc object torqu produc chang rotat speed object extend bodi part usual appli forc adjac part distribut forc bodi intern mec
han stress intern mechan stress caus acceler bodi forc balanc one anoth pressur distribut mani small forc appli area bodi simpl type
stress unbalanc caus bodi acceler stress usual caus deform solid materi flow fluid
[9.999999999999998e-15, 9.999999999999998e-15, 9.999999999999998e-15, 9.999999999999998e-15, 9.999999999999998e-15, 9.999999999999998e-15]
```

```
query is: forc
relevant doc: physic forc influenc chang motion object forc caus object mass chang veloc e g move state rest e acceler forc also des
crib intuit push pull forc ha magnitud direct make vector quantiti measur si unit newton n forc repres symbol f former p origin form
newton second law state net forc act upon object equal rate momentum chang time mass object constant law impli acceler object direct
proport net forc act object direct net forc invers proport mass object concept relat forc includ thrust increas veloc object drag de
creas veloc object torqu produc chang rotat speed object extend bodi part usual appli forc adjac part distribut forc bodi intern mec
han stress intern mechan stress caus acceler bodi forc balanc one anoth pressur distribut mani small forc appli area bodi simpl type
stress unbalanc caus bodi acceler stress usual caus deform solid materi flow fluid
[1, 1, 1, 1, 1, 1]
```

```
query is: thermodynam
relevant doc: physic forc influenc chang motion object forc caus object mass chang veloc e g move state rest e acceler forc also des
crib intuit push pull forc ha magnitud direct make vector quantiti measur si unit newton n forc repres symbol f former p origin form
newton second law state net forc act upon object equal rate momentum chang time mass object constant law impli acceler object direct
proport net forc act object direct net forc invers proport mass object concept relat forc includ thrust increas veloc object drag de
creas veloc object torqu produc chang rotat speed object extend bodi part usual appli forc adjac part distribut forc bodi intern mec
han stress intern mechan stress caus acceler bodi forc balanc one anoth pressur distribut mani small forc appli area bodi simpl type
stress unbalanc caus bodi acceler stress usual caus deform solid materi flow fluid
[1, 1, 1, 1, 1, 1]
```

# Output

query is: world war 2 histor event

relevant doc: world war ii second world war often abbrevi wwii ww2 wa global war last 1939 1945 involv vast major world countriesâ i nclud great powersâ form two oppos militari allianc alli axi power total war direct involv 100 million personnel 30 countri major pa rticip threw entir econom industri scientif capabl behind war effort blur distinct civilian militari resourc aircraft play major rol e conflict enabl strateg bomb popul centr two us nuclear weapon war day world war ii wa far deadliest conflict human histori result 70 85 million fatal major civilian ten million peopl die due genocid includ holocaust starvat massacr diseas wake axi defeat germani japan occupi war crime tribun conduct german japanes leader

[9.9999999999999996e-29, 9.9999999999999996e-29, 9.9999999999999996e-29, 9.9999999999999996e-29, 2.678581428571428e-23, 6.849415068493149e-24]

query is: nuclear bomb

relevant doc: nuclear weapon also known atom bomb atom bomb nuclear bomb nuclear warhead colloqui bomb nuke explos devic deriv destr uct forc nuclear reaction either fission fission bomb combin fission fusion reaction thermonuclear bomb bomb type releas larg quanti ti energi relat small amount matter first test fission atom bomb releas amount energi approxim equal 20 000 ton tnt 84 tj 1 first th ermonuclear hydrogen bomb test releas energi approxim equal 10 million ton tnt 42 pj nuclear bomb yield 10 ton tnt w54 50 megaton ts ar bomba see tnt equival thermonuclear weapon weigh littl 2 400 pound 1 100 kg releas energi equal 1 2 million ton tnt 5 0 pj 2 nucl ear devic larger convent bomb devast entir citi blast fire radiat sinc weapon mass destruct prolifer nuclear weapon focus intern rel at polici nuclear weapon deploy twice war unit state japanes citi hiroshima nagasaki 1945 world war ii

[1e-07, 1e-07, 1e-07, 1e-07, 1e-07, 0.0136987301369863]

# Conclusion

- ❖ We have seen in this project the results of our application model, so we conclude from that the Unigram for these queries is much better than the use of Bigram.
- ❖ We also see that not all the results for Bigram are correct, knowing that writing the code and the query are correct, which means that spars diagram probability occurs.



**Thank you!**