NATURAL LANGUAGE PROCESSING

AGENDA

DEFINITION

WHY NLP

NLP TASK

TOOLS AND APPROACH

NLP USECASES

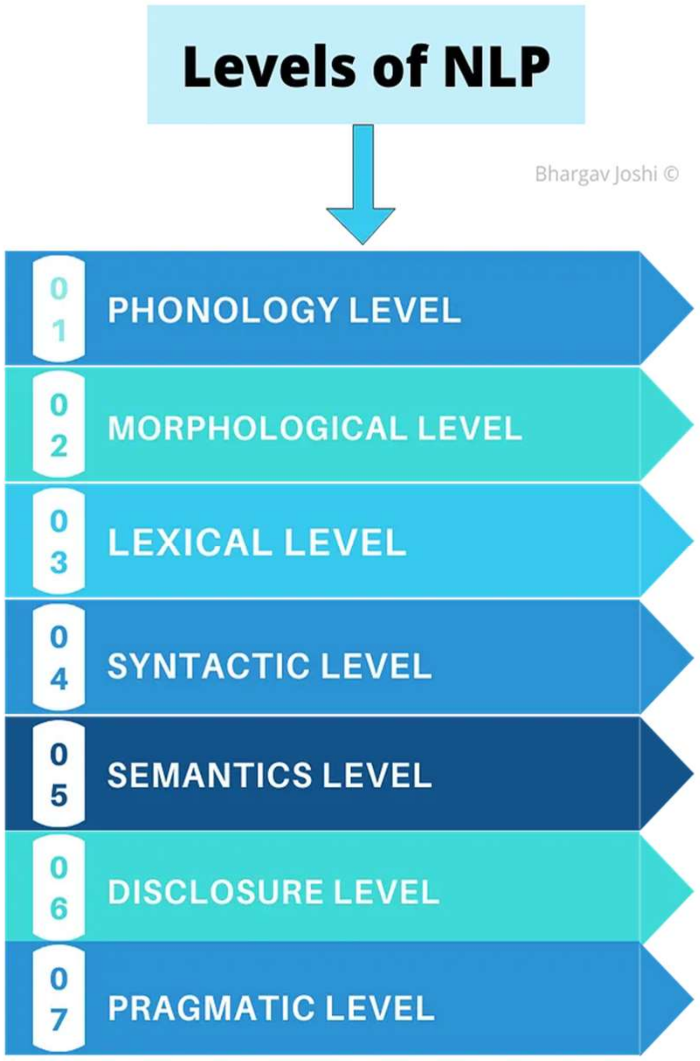
DEFINITION

Natural language processing (NLP) refers to the branch of computer science—and more specifically, the branch of artificial intelligence or AI—concerned with giving computers the ability to understand text and spoken words in much the same way human beings can.

NLP combines computational linguistics—rule-based modeling of human language—with statistical, machine learning, and deep learning models. Together, these technologies enable computers to process human language in the form of text or voice data and to 'understand' its full meaning, complete with the speaker or writer's intent and sentiment.

NLP CHALLENGES

NLP faces various challenges, including language ambiguity, understanding context, handling different languages, and training models on large amounts of data. Additionally, ethical considerations such as bias and privacy need to be addressed in NLP application.



- At this level basically, it deals with pronunciation.
- It deals with the interpretation of speech sound across words
- It deals with the smallest words that convey meaning and suffixes and prefixes.
- This deals with the study at the level of words with respect to their lexical meaning and Part of speech (POS)
- This level deals with the grammar and structure of sentences
- This level deals with the meaning of words and sentences
- This level deals with the structure of different kinds of text.
- This level deals with the use real world knowledge and understanding of how this influences the meaning of what is being communicated.

NLP TASK

Some of these tasks include the following:

- **Speech recognition**, also called speech-to-text, is the task of reliably converting voice data into text data. Speech recognition is required for any application that follows voice commands or answers spoken questions. What makes speech recognition especially challenging is the way people talk—quickly, slurring words together, with varying emphasis and intonation, in different accents, and often using incorrect grammar.
- **Part of speech tagging**, also called grammatical tagging, is the process of determining the part of speech of a particular word or piece of text based on its use and context. Part of speech identifies 'make' as a verb in 'I can make a paper plane,' and as a noun in 'What make of car do you own?'

NLP TASK (1)

- Word sense disambiguation is the selection of the meaning of a word with multiple meanings through a process of semantic analysis that determine the word that makes the most sense in the given context. For example, word sense disambiguation helps distinguish the meaning of the verb 'make' in ‘make the grade’ (achieve) vs. ‘make a bet’ (place).
- Named entity recognition, or NEM, identifies words or phrases as useful entities. NEM identifies ‘Kentucky’ as a location or ‘Fred’ as a man's name.
- Co-reference resolution is the task of identifying if and when two words refer to the same entity. The most common example is determining the person or object to which a certain pronoun refers (e.g., ‘she’ = ‘Mary’), but it can also involve identifying a metaphor or an idiom in the text (e.g., an instance in which 'bear' isn't an animal but a large hairy person).

NLP TASK (2)

- **Sentiment analysis** attempts to extract subjective qualities—attitudes, emotions, sarcasm, confusion, suspicion—from text.
- **Natural language generation** is sometimes described as the opposite of speech recognition or speech-to-text; it's the task of putting structured information into human language.

NLP TOOLS & APPROACHES

- Python and the natural language toolkit (NLTK)

The Python programing language provides a wide range of tools and libraries for attacking specific NLP tasks. Many of these are found in the Natural Language Toolkit, or NLTK, an open source collection of libraries, programs, and education resources for building NLP programs.

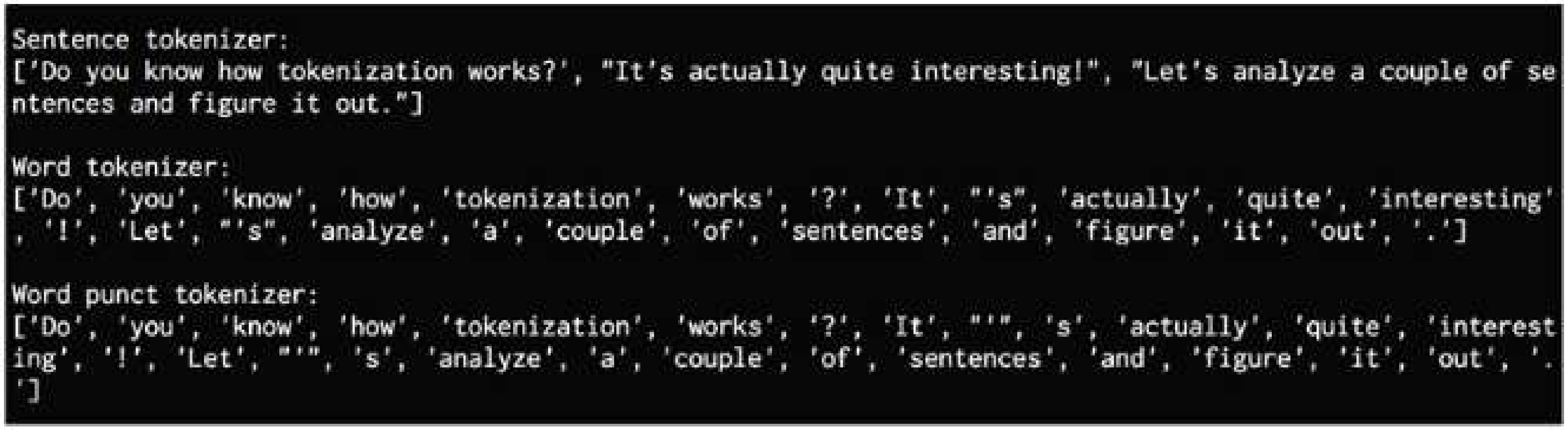
You can find more information about NLTK at https://www.nltk.org/

TEXT PROCESSING

- Tokenization, process of dividing text into individual units call tokens.these tokens can be words, phrases, or characters depending on the specific task at hand.
- Stop words removal, stopwords are common words that do not carry much meaning, such us “and”, “the”, or “is”. This procedure helps reduce noise in the text.
- Stemming, reducing words to their base or root form kown as the stem
- Lemmatization, similar to stemming, but it aims to reduce words to their dictionary or canonical form, known as the lemma.

TOKENIZING TEXT DATA

When we deal with text, we need to break it down into smaller pieces for analysis. To do this, tokenization can be applied. Tokenization is the process of dividing text into a set of pieces, such as words or sentences. These pieces are called tokens. Depending on what we want to do, we can define our own methods to divide the text into many tokens. Let's look at how to tokenize the input text using NLTK.

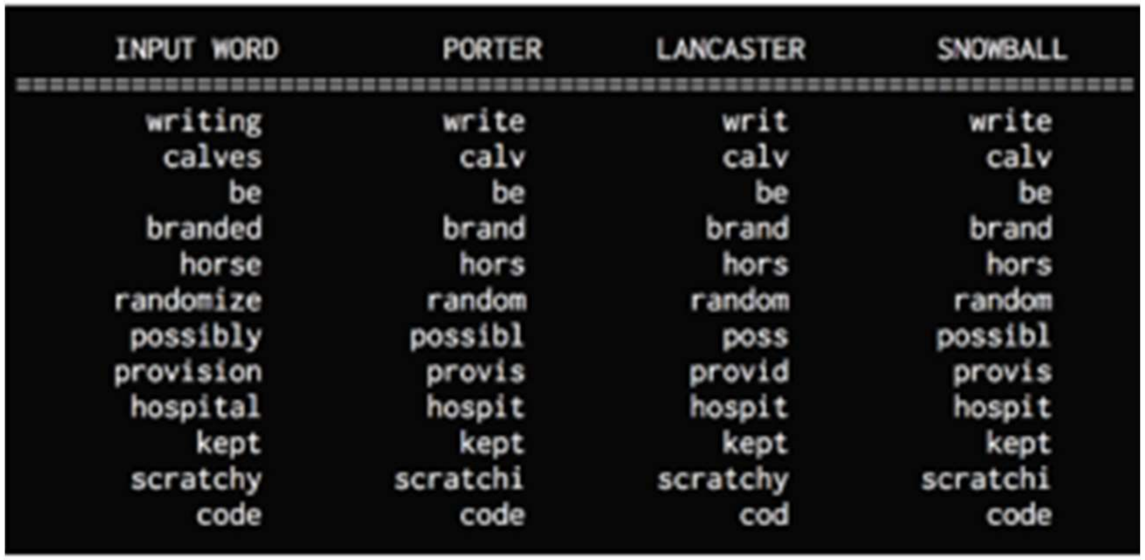


```
Sentence tokenizer:
['Do you know how tokenization works?', "It's actually quite interesting!", "Let's analyze a couple of se
ntences and figure it out."]

Word tokenizer:
['Do', 'you', 'know', 'how', 'tokenization', 'works', '?', 'It', "'s", 'actually', 'quite', 'interesting'
, '!', 'Let', "'s", 'analyze', 'a', 'couple', 'of', 'sentences', 'and', 'figure', 'it', 'out', '.']

Word punct tokenizer:
['Do', 'you', 'know', 'how', 'tokenization', 'works', '?', 'It', "'", 's', 'actually', 'quite', 'interest
ing', '!', 'Let', "'", 's', 'analyze', 'a', 'couple', 'of', 'sentences', 'and', 'figure', 'it', 'out', '.
']
```

STEMMING

Working with text means working with a lot of variation. We must deal with different forms of the same word and enable the computer to understand that these different words have the same base form. For example, singer, singing, song, sung, and so on. This set of words share similar meanings.

Stemming is a way of producing morphological variants of a root/base word. Humans can easily identify these base forms and derive context. When analyzing text, it's useful to extract these base forms. It is basically a heuristic process that cuts off the ends of words to extract their base forms. Let's see how to do it using NLTK.

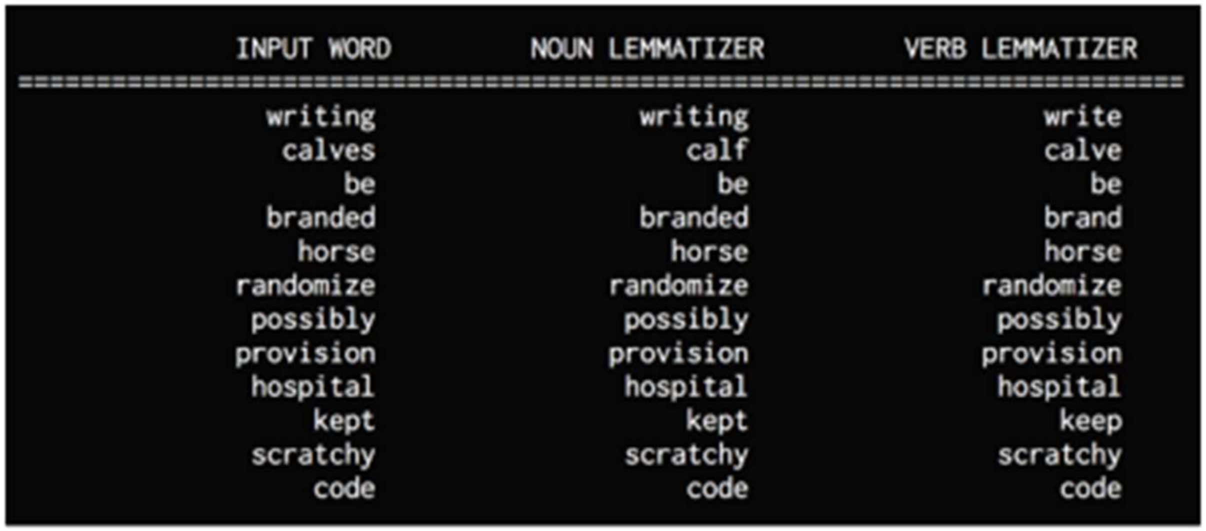


```
   INPUT WORD         PORTER      LANCASTER       SNOWBALL
==================================================================
      writing          write           writ          write
      calves           calv            calv           calv
           be             be             be             be
     branded          brand          brand          brand
       horse           hors           hors           hors
   randomize         random         random         random
    possibly        possibl           poss        possibl
   provision         provis         provid         provis
    hospital         hospit         hospit         hospit
        kept           kept           kept           kept
    scratchy        scratchi       scratchy       scratchi
        code           code            cod           code
```

LEMMATIZATION

Lemmatization is another method of reducing words to their base forms. In the revious section, we saw that some of the base forms that were obtained from those stemmers didn't make sense. Lemmatization is like stemming, but it brings context to the words. So, it links words with similar meanings to one word.

The lemmatization process uses the lexical and morphological analysis of words. It obtains the base forms by removing the inflectional word endings such as ing or ed. This base form of any word is known as the lemma. If you lemmatize the word calves, you should get calf as the output. One thing to note is that the output depends on whether the word is a verb or a noun. Let's look at how to do this with NLTK.

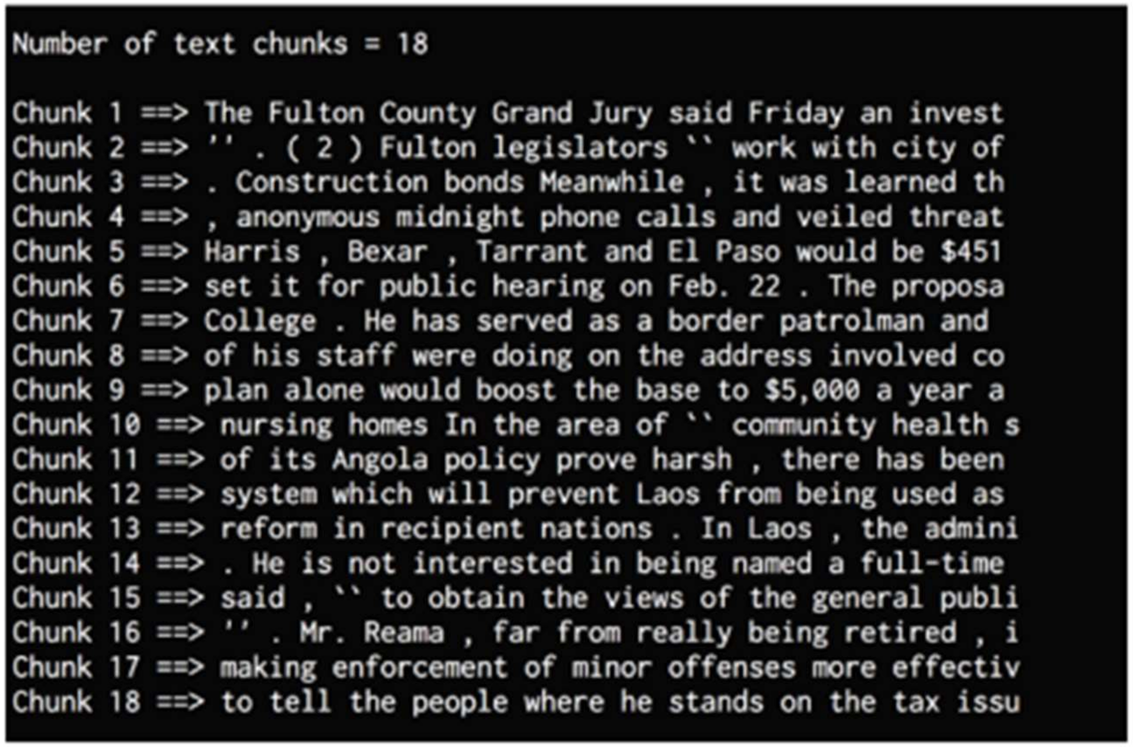


```
          INPUT WORD          NOUN LEMMATIZER          VERB LEMMATIZER
==========================================================================
             writing                  writing                    write
              calves                     calf                    calve
                  be                       be                       be
             branded                  branded                    brand
               horse                    horse                    horse
           randomize                randomize                randomize
            possibly                 possibly                 possibly
           provision                provision                provision
            hospital                 hospital                 hospital
                kept                     kept                     keep
            scratchy                 scratchy                 scratchy
                code                     code                     code
```

DIVIDING TEXT DATA INTO CHUNKS

Text data usually needs to be divided into pieces for further analysis. This process is known as chunking. This is used frequently in text analysis. The conditions that are used to divide the text into chunks can vary based on the problem at hand.

This is not the same as tokenization, where text is also divided into pieces. During chunking, we do not adhere to any constraints, except for the fact that the output chunks need to be meaningful. When we deal with large text documents, it becomes important to divide the text into chunks to extract meaningful information.

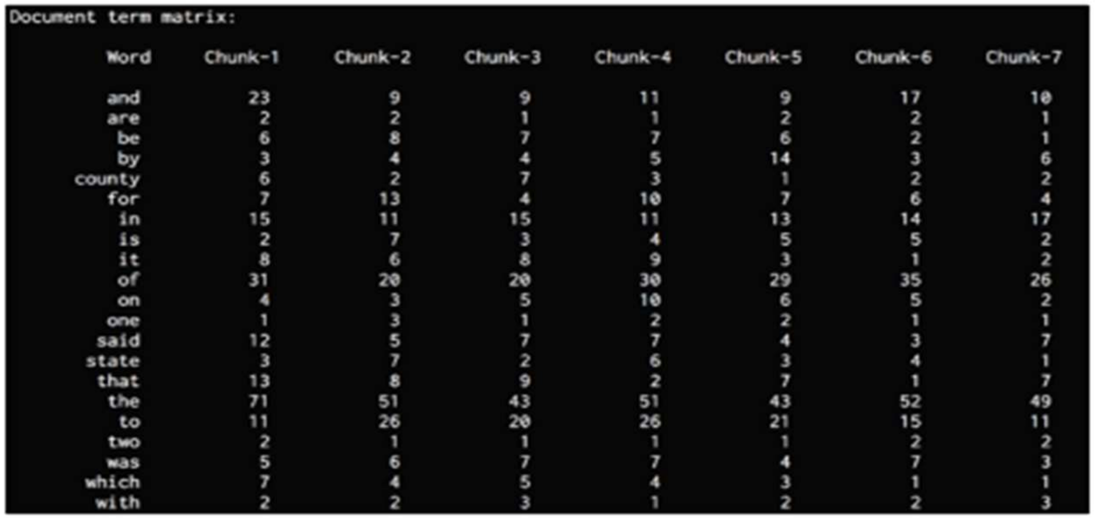


```
Number of text chunks = 18

Chunk 1 ==> The Fulton County Grand Jury said Friday an invest
Chunk 2 ==> '' . ( 2 ) Fulton legislators `` work with city of
Chunk 3 ==> . Construction bonds Meanwhile , it was learned th
Chunk 4 ==> , anonymous midnight phone calls and veiled threat
Chunk 5 ==> Harris , Bexar , Tarrant and El Paso would be $451
Chunk 6 ==> set it for public hearing on Feb. 22 . The proposa
Chunk 7 ==> College . He has served as a border patrolman and
Chunk 8 ==> of his staff were doing on the address involved co
Chunk 9 ==> plan alone would boost the base to $5,000 a year a
Chunk 10 ==> nursing homes In the area of `` community health s
Chunk 11 ==> of its Angola policy prove harsh , there has been
Chunk 12 ==> system which will prevent Laos from being used as
Chunk 13 ==> reform in recipient nations . In Laos , the admini
Chunk 14 ==> . He is not interested in being named a full-time
Chunk 15 ==> said , `` to obtain the views of the general publi
Chunk 16 ==> '' . Mr. Reama , far from really being retired , i
Chunk 17 ==> making enforcement of minor offenses more effectiv
Chunk 18 ==> to tell the people where he stands on the tax issu
```

EXTRACTING THE FREQUENCY OF TERMS USING THE BAG OF WORDS MODEL

One of the main goals of text analysis with the Bag of Words model is to convert text into a numerical form so that we can use machine learning on it. Let's consider text documents that contain many millions of words. In order to analyze these documents, we need to extract the text and convert it into a form of numerical representation.

Machine learning algorithms need numerical data to work with so that they can analyze the data and extract meaningful information. This is where the Bag of Words model comes in. This model extracts vocabulary from all the words in the documents and builds a model using a document-term matrix. This allows us to represent every document as a bag of words. We just keep track of word counts and disregard the grammatical details and the word order.

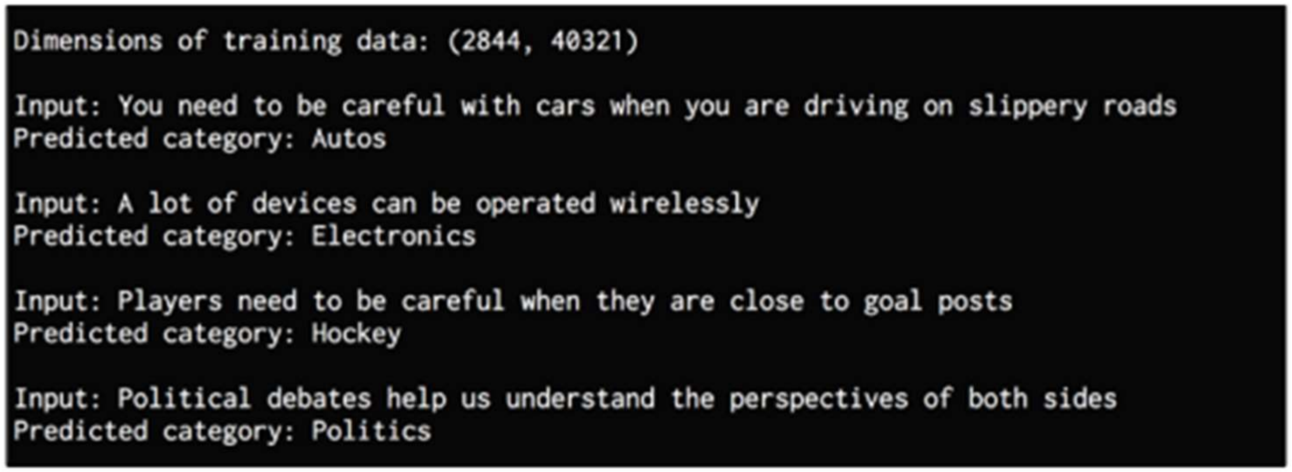


```
Document term matrix:

      Word    Chunk-1    Chunk-2    Chunk-3    Chunk-4    Chunk-5    Chunk-6    Chunk-7

       and         23          9          9         11          9         17         10
       are          2          2          1          1          2          2          1
        be          6          8          7          7          6          2          1
        by          3          4          4          5         14          3          6
    county          6          2          7          3          1          2          2
       for          7         13          4         10          7          6          4
        in         15         11         15         11         13         14         17
        is          2          7          3          4          5          5          2
        it          8          6          8          9          3          1          2
        of         31         20         20         30         29         35         26
        on          4          3          5         10          6          5          2
       one          1          3          1          2          2          1          1
      said         12          5          7          7          4          3          7
     state          3          7          2          6          3          4          1
      that         13          8          9          2          7          1          7
       the         71         51         43         51         43         52         49
        to         11         26         20         26         21         15         11
       two          2          1          1          1          1          2          2
       was          5          6          7          7          4          7          3
     which          7          4          5          4          3          1          1
      with          2          2          3          1          2          2          3
```

BUILDING A CATEGORY PREDICTOR

A category predictor is used to predict the category to which a given piece of text belongs. This is frequently used in text classification to categorize text documents. Search engines frequently use this tool to order search results by relevance. For example, let's say that we want to predict whether a given sentence belongs to sports, politics, or science. To do this, we build a corpus of data and train an algorithm. This algorithm can then be used

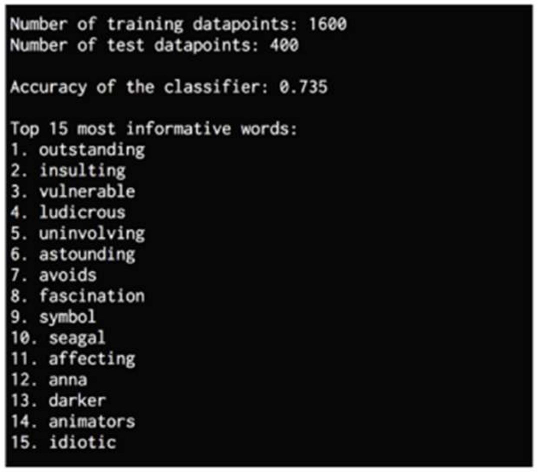
In order to build this predictor, we will use a metric called Term Frequency – Inverse Document Frequency (tf-idf). In a set of documents, we need to understand the importance of each word. The tf-idf metric helps us to understand how important a given word is to a document in a set of documents. for inference on unknown data.



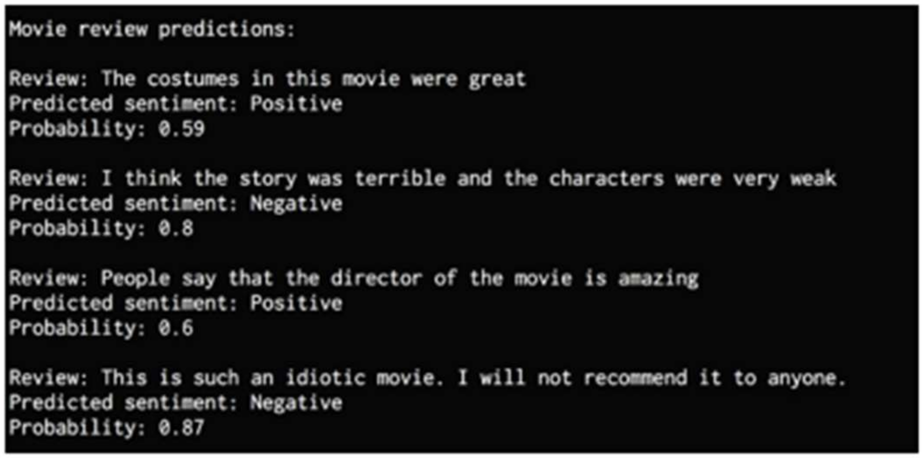
```
Dimensions of training data: (2844, 40321)

Input: You need to be careful with cars when you are driving on slippery roads
Predicted category: Autos

Input: A lot of devices can be operated wirelessly
Predicted category: Electronics

Input: Players need to be careful when they are close to goal posts
Predicted category: Hockey

Input: Political debates help us understand the perspectives of both sides
Predicted category: Politics
```

BUILDING A SENTIMENT ANALYZER

Sentiment analysis is the process of determining the sentiment of a piece of text. For example, it can be used to determine whether a movie review is positive or negative. This is one of the most popular applications of natural language processing. We can add more categories as well, depending on the problem at hand. This technique can be used to get a sense of how people feel about a product, brand, or topic. It is frequently used to analyze marketing campaigns, opinion polls, social media presence, product reviews on e-commerce sites, and so on. Let's see how to determine the sentiment of a movie review. We will use a Naive Bayes classifier to build this sentiment analyzer. First, extract all the unique words from the text. The NLTK classifier needs this data to be arranged in the form of a dictionary so that it can ingest it. Once the text data is divided into training and testing datasets, the Naive Bayes classifier will be trained to classify the reviews into positive and negative. Afterward, the top most informative words to indicate positive and negative reviews can be calculated and displayed. This information is interesting because it shows what words are being used to denote various reactions.



```
Number of training datapoints: 1600
Number of test datapoints: 400

Accuracy of the classifier: 0.735

Top 15 most informative words:
1. outstanding
2. insulting
3. vulnerable
4. ludicrous
5. uninvolving
6. astounding
7. avoids
8. fascination
9. symbol
10. seagal
11. affecting
12. anna
13. darker
14. animators
15. idiotic
```



```
Movie review predictions:

Review: The costumes in this movie were great
Predicted sentiment: Positive
Probability: 0.59

Review: I think the story was terrible and the characters were very weak
Predicted sentiment: Negative
Probability: 0.8

Review: People say that the director of the movie is amazing
Predicted sentiment: Positive
Probability: 0.6

Review: This is such an idiotic movie. I will not recommend it to anyone.
Predicted sentiment: Negative
Probability: 0.87
```

THANK YOU