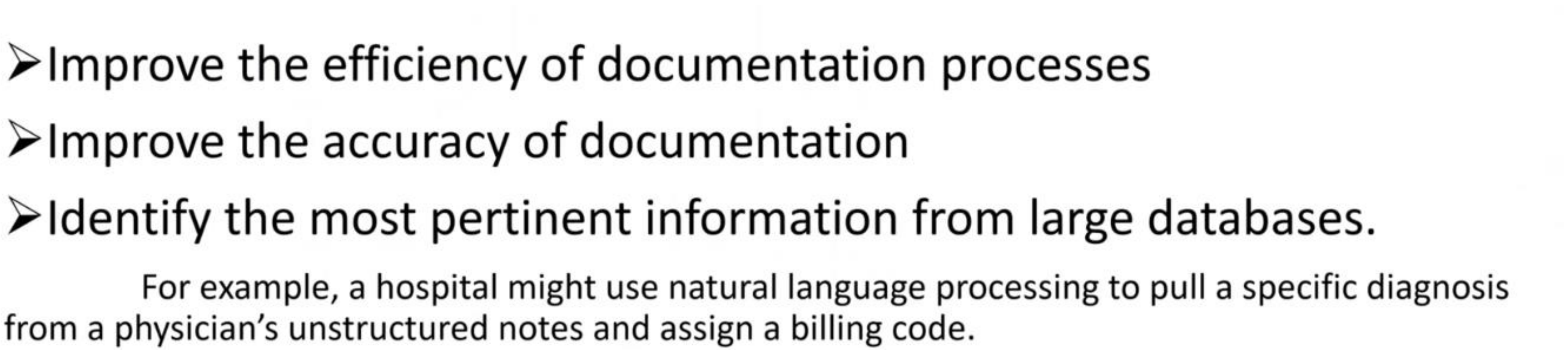
Natural Language Processing

BY MG ANALYTICS

NLP - Advantages

- The benefits of natural language processing are innumerable.
- Natural language processing can be leveraged by companies to



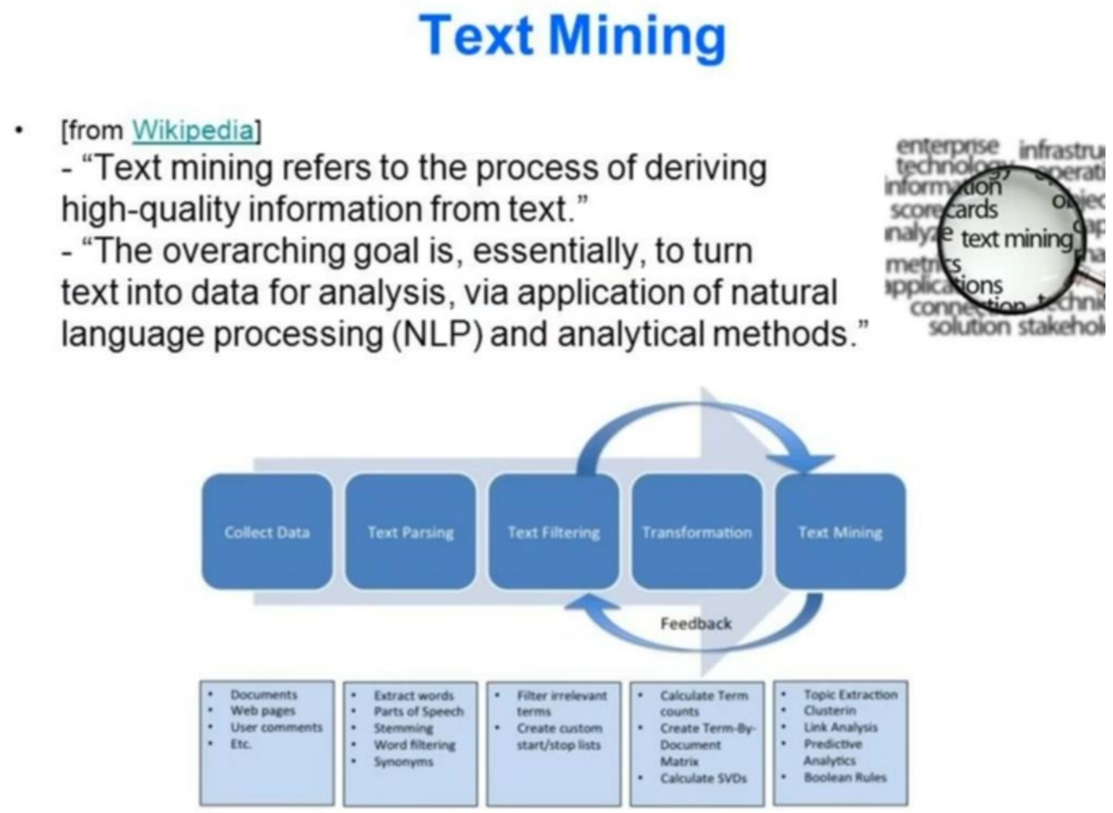
➢Improve the efficiency of documentation processes

➢Improve the accuracy of documentation

➢Identify the most pertinent information from large databases.

For example, a hospital might use natural language processing to pull a specific diagnosis from a physician's unstructured notes and assign a billing code.

Text Mining

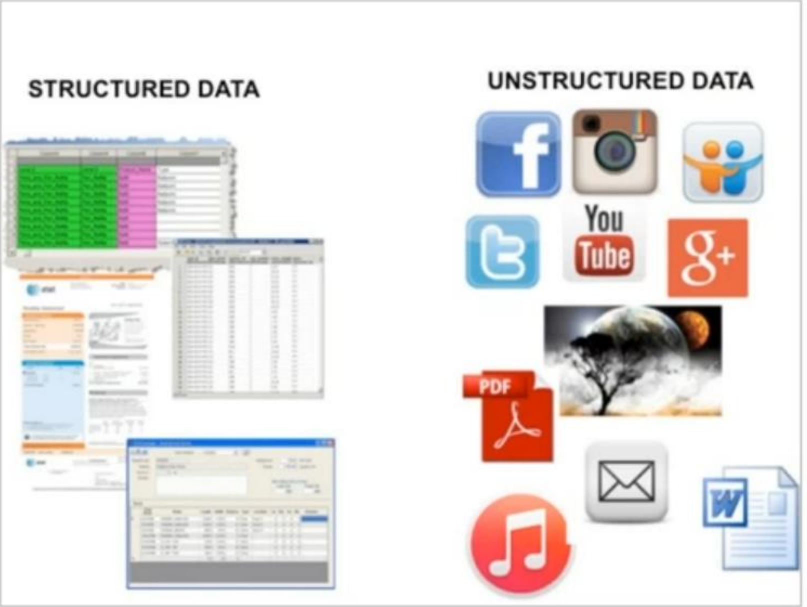


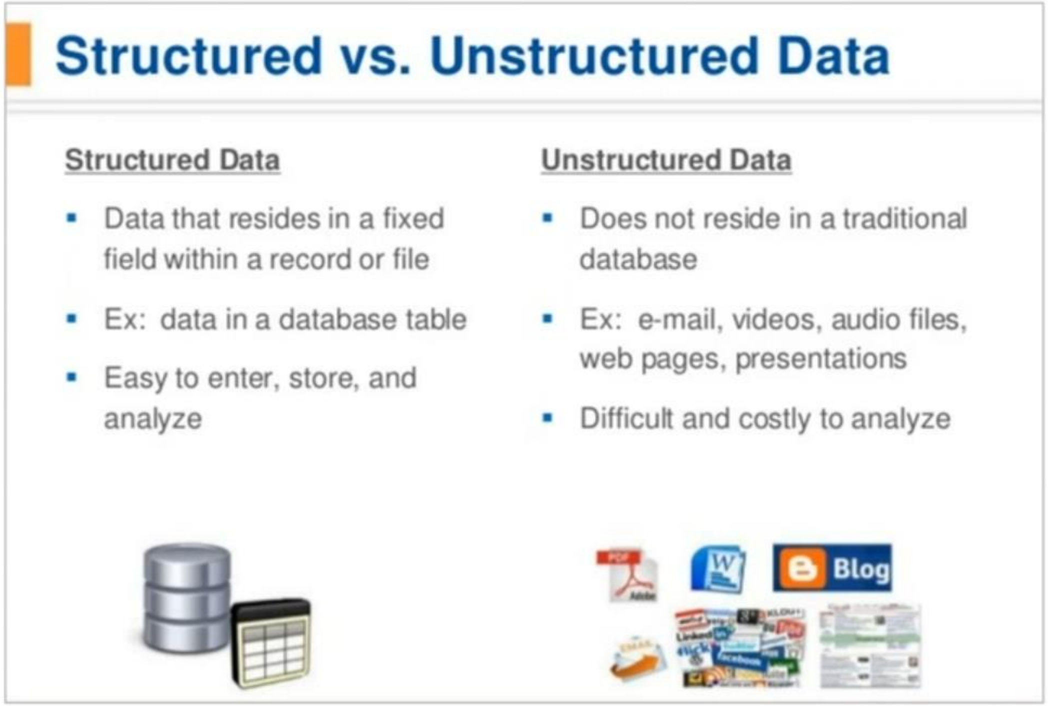
Text Mining

- [from Wikipedia]
  - "Text mining refers to the process of deriving high-quality information from text."
  - "The overarching goal is, essentially, to turn text into data for analysis, via application of natural language processing (NLP) and analytical methods."

Collect Data	Text Parsing	Text Filtering	Transformation	Text Mining
• Documents • Web pages • User comments • Etc.	• Extract words • Parts of Speech • Stemming • Word filtering • Synonyms	• Filter irrelevant terms • Create custom start/stop lists	• Calculate Term counts • Create Term-By-Document Matrix • Calculate SVDs	• Topic Extraction • Clusterin • Link Analysis • Predictive Analytics • Boolean Rules

Structured Data Vs Unstructured Data





Structured vs. Unstructured Data

Structured Data

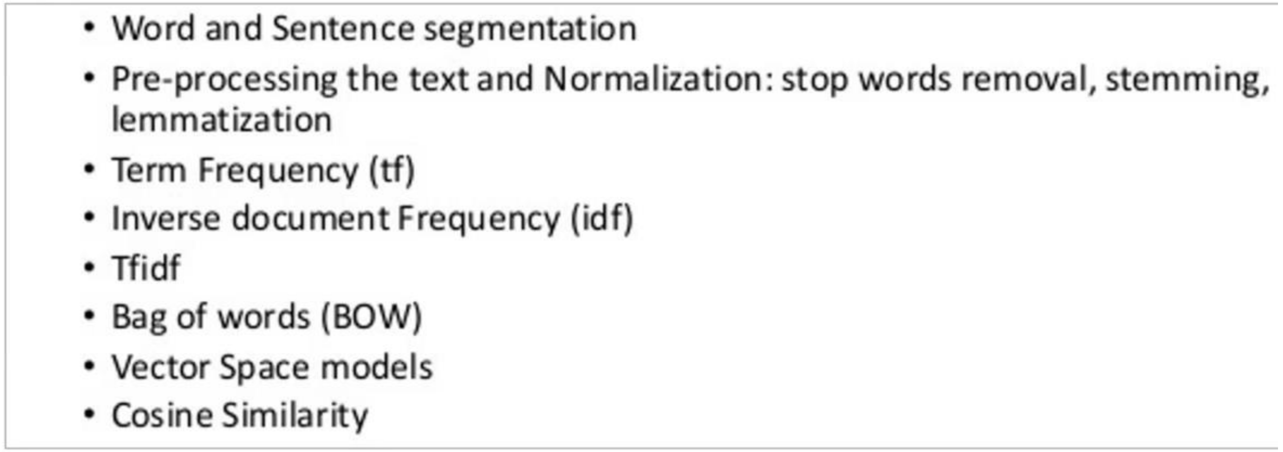
- Data that resides in a fixed field within a record or file
- Ex: data in a database table
- Easy to enter, store, and analyze

Unstructured Data

- Does not reside in a traditional database
- Ex: e-mail, videos, audio files, web pages, presentations
- Difficult and costly to analyze

NLP – Basic Concepts & Terms

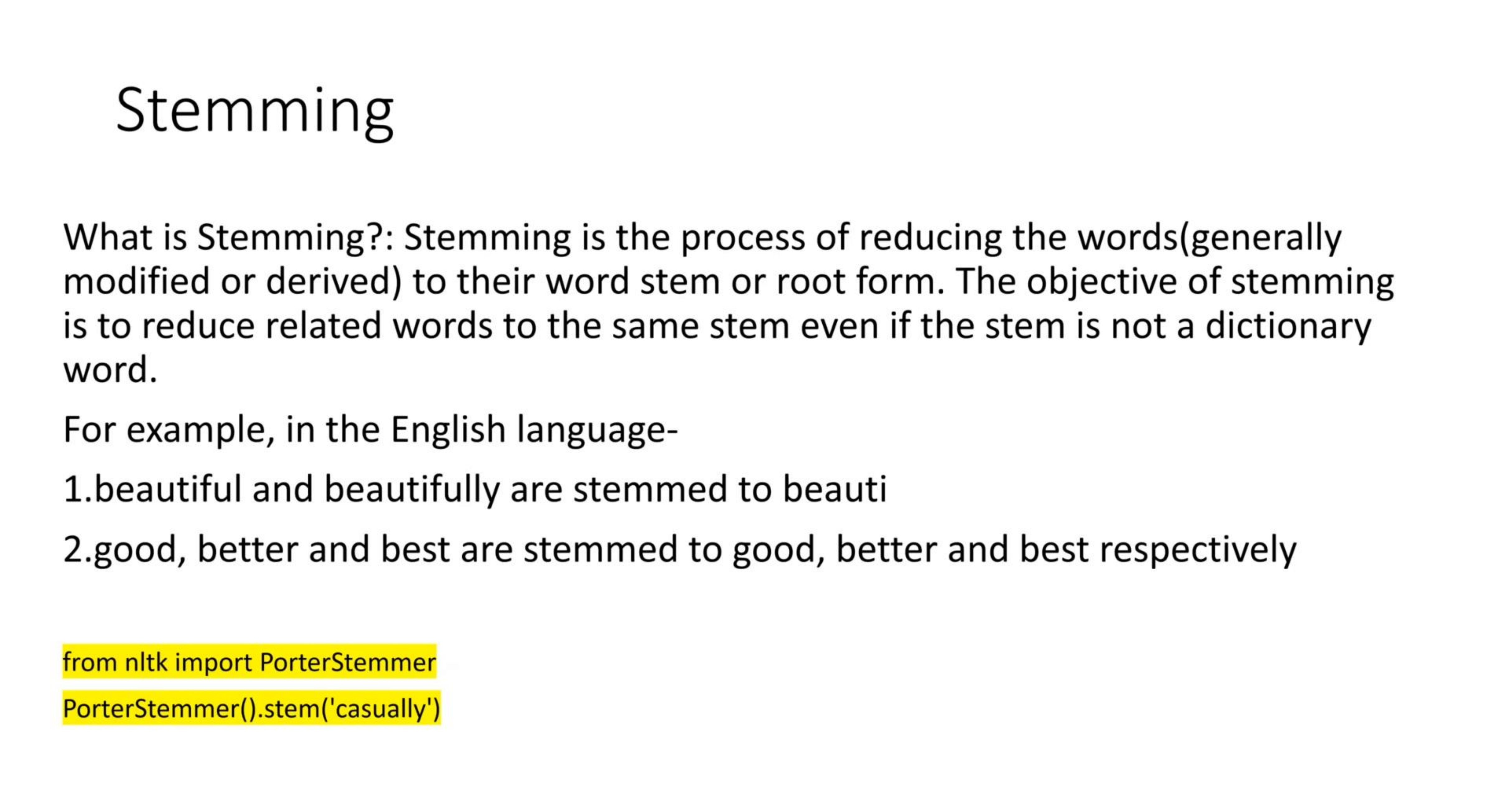
- Tokenization – process of converting a text into tokens
- Tokens – words or entities present in the text
- Text object – a sentence or a phrase or a word or an article



- Word and Sentence segmentation
- Pre-processing the text and Normalization: stop words removal, stemming, lemmatization
- Term Frequency (tf)
- Inverse document Frequency (idf)
- Tfidf
- Bag of words (BOW)
- Vector Space models
- Cosine Similarity

my “red-blue” socks are the prettiest socks in the country,, no other, red blue socks are prettier in the nation….

my red blue socks prettiest socks country no other red blue socks prettier nation



Stemming

What is Stemming?: Stemming is the process of reducing the words(generally modified or derived) to their word stem or root form. The objective of stemming is to reduce related words to the same stem even if the stem is not a dictionary word.

For example, in the English language-

1.beautiful and beautifully are stemmed to beauti

2.good, better and best are stemmed to good, better and best respectively

```
from nltk import PorterStemmer
PorterStemmer().stem('casually')
```

my red blue socks prettiest socks country no other red blue socks prettier nation

my red blue socks pretti socks country no other red blue socks pretti nation

Lemmatization

- The process of reducing a group of words into their lemma or dictior form. It takes into account things like POS(Parts of Speech), the mear of the word in the sentence, the meaning of the word in the nearby sentences etc. before reducing the word to its lemma.

For example, in the English Language- beautiful and beautifully are lemmatized to beautiful and beautifully respectively.
good, better and best are lemmatized to good, good and good respect

```
from nltk.stem import WordNetLemmatizer
lemmatizer = WordNetLemmatizer()
print(lemmatizer.lemmatize("better"))#, pos="a"))
print(lemmatizer.lemmatize("better", pos="a"))
```

my red blue socks prettiest socks country
no other red blue socks prettier nation

my red blue socks pretty socks country no
other red blue socks pretty country

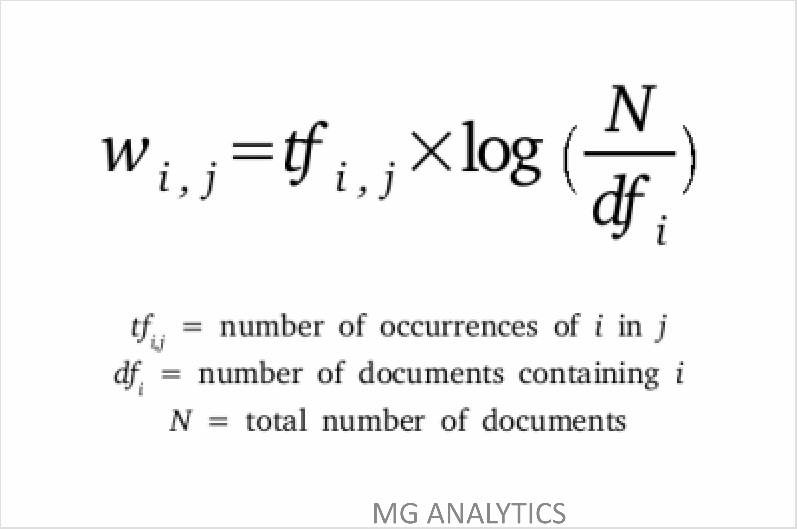
Lemmatization and Stemming

- **Lemmatization** is closely related to **stemming**. The difference is that a stemmer operates on a single word without knowledge of the context, and therefore cannot discriminate between words which have different meanings depending on part of speech. However, stemmers are typically easier to implement and run faster, and the reduced accuracy may not matter for some applications.
- A **stemmer** will return the stem of a word, which needn't be identical to the morphological root of the word. It usually sufficient that related words map to the same stem, even if the stem is not in itself a valid root, while in **lemmatisation**, it will return the dictionary form of a word, which must be a valid word.
- In **lemmatisation**, the part of speech of a word should be first determined and the normalisation rules will be different for different part of speech, while the **stemmer** operates on a single word without knowledge of the context, and therefore cannot discriminate between words which have different meanings depending on part of speech.

Statistical Features

Text data can also be quantified directly into numbers using several techniques described in this section:

- TF-IDF (Term Frequency – Inverse Document Frequency)
- Count Features



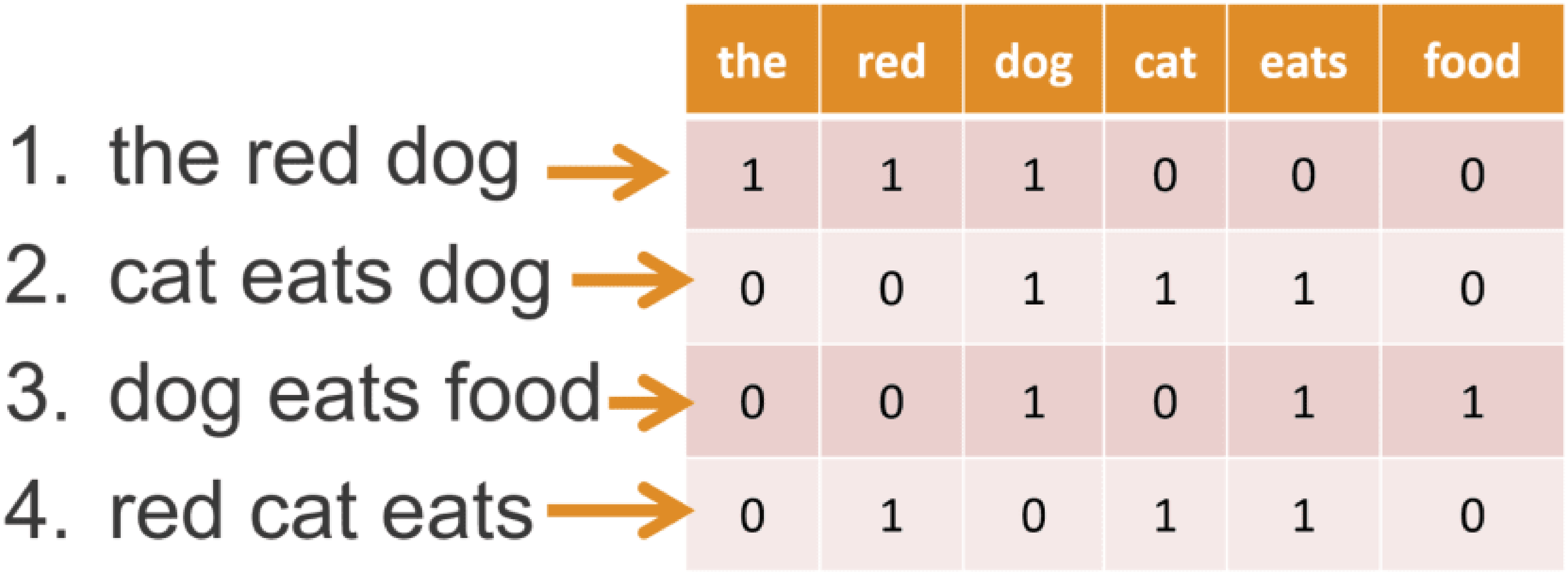
$$w_{i,j} = tf_{i,j} \times \log\left(\frac{N}{df_i}\right)$$

$tf_{i,j}$ = number of occurrences of i in j

df_i = number of documents containing i

N = total number of documents

my	red	blue	socks	pretty	country	no	other
1	2	2	2	2	2	1	1



1. the red dog
2. cat eats dog
3. dog eats food
4. red cat eats

the	red	dog	cat	eats	food
1	1	1	0	0	0
0	0	1	1	1	0
0	0	1	0	1	1
0	1	0	1	1	0

Code Walk through