working with text

Calculating Word Importance

TF-IDF answers the question: How important is each word? (or how important is each word stem).

TF-IDF is a key factor utilized in search engines.

TF is **Term Frequency**; the number of times the word occurs.

- It is logical to draw correlation between frequency and importance.
- However, this is not the case → "The" is a very frequent word, but holds little importance.
- Term frequency cannot be used on its own to determine word importance

IDF is **Inverse Document Frequency**:

- log(1/(fraction of documents the word appears in))

TF_IDF = Term Frequency X Inverse Document Frequency

Therefore, when is TF_IDF **high**?

- When the term appears **many** times in **few** documents

When is TF_IDF **low**?

- When the term appears in almost all documents
- When the term does not appear often

$$\text{TF} \cdot \log\left(\frac{\#Documents}{\#Documents\ the\ Word\ Appears}\right)$$

Natural Language Processing

There are many subfields in NLP, some major ones are discussed as follows:

Named Entity Recognition

The goal of NER is to label words that are names of things:

people, organizations, locations, gene proteins, etc.

For example, take the following sentence:

" Cynthia and Steve worked for Duke and Quantia Analytics in 2016 in Niagara .

Person
Organization
Location

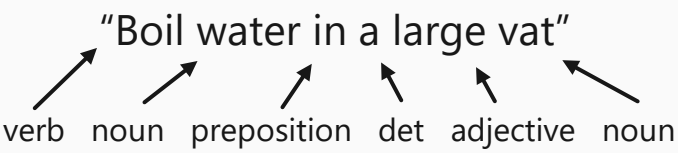
NER analysis is useful for answering questions such as:
What are all organizations mentioned in a set of legal documents?
What places were involved in articles about a military group?

Part of Speech Tagging

Another subfield of NLP is Part of Speech Tagging, where the goal is to assign each word in a sentence to a part of speech:

noun, adjective, verb, adverb

For example, take the following sentence:



The technique is difficult in applying to machine learning because of the double entendre in English:

For example, *water* is a **noun** in the sentence above but could also be used as **verb** as follows:

> "**Water** the plant"

The importance behind this distinction is illustrated while building a speech synthesizer that needs to sound natural:

When text becomes speech, the noun version of a word is sometimes pronounced differently than the verb version.

Take for example:

> "I **object** to your taking this class"
>
> "What is that **object** you have in your hand?"

Probabilistic machine learning models are used to do the tagging:

> Sentence $W = w_1, w_2, w_3, \dots, w_n$ "Boil water in a large vat"
>
> Assign a sequence of tags $T = t_1, t_2, t_3, \dots, t_n$

Choose T to maximize $P(T|W)$: Maximize the probability that the words arise from a tag.