NLP Lecture 1

15th July 2021 -- MTECH

About Me

Ekta Shah

- Senior Machine Learning Engineer at Quantiphi
- Data Scientist at Viacom18
- Data Science Intern at General Mills
- Software Developer at BNP Paribas
- Passion for Teaching

Agenda for Today

- Course Overview - Syllabus
- Introduction to NLP
- Machine Learning and NLP
- Argmax function
- Syntactic Collocation
- Data Pipeline

Introduction to NLP

- **Wiki: Natural language processing** (**NLP**) is a field of computer science, artificial intelligence, and computational linguistics concerned with the interactions between computers and human (**natural**) languages.

Go beyond the keyword matching

- Identify the structure and meaning of words, sentences, texts and conversations
- Deep understanding of broad language
- NLP is all around us

Why is NLP interesting?

Languages involve many human activities− Reading, writing, speaking, listening

- Voice can be used as an user interface in many

applications− Remote controls, virtual assistants like siri,...

- NLP is used to acquire insights from massive

amount of textual data− E.g., hypotheses from medical, health reports

- NLP has many applications
- NLP is hard!

Why is NLP hard?

I shot an elephant in my pajamas.

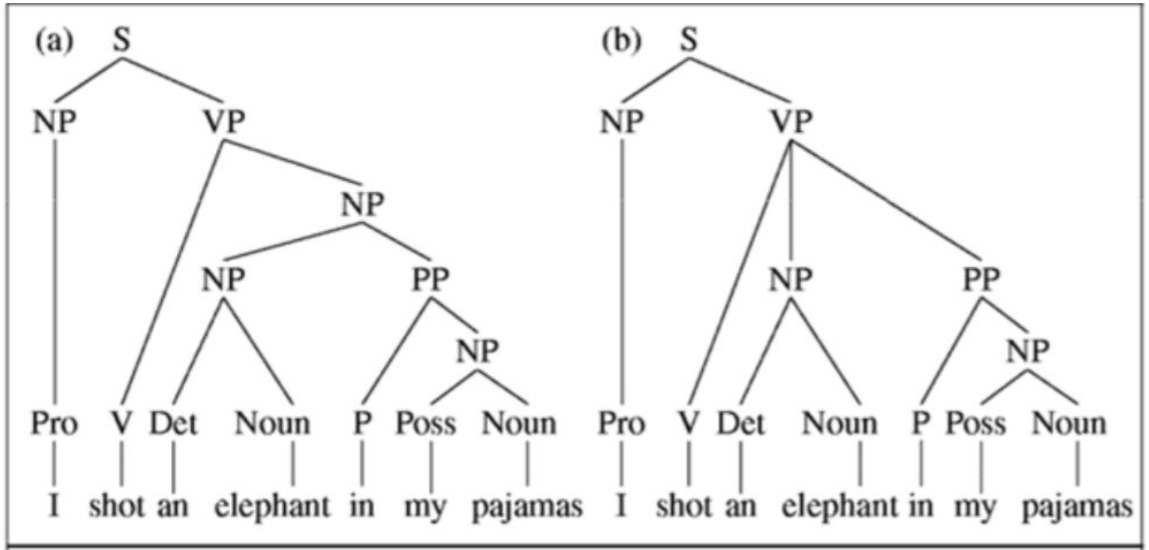


Figure 10.11 Two parse trees for an ambiguous sentence. Parse (a) corresponds to the humorous reading in which the elephant is in the pajamas, parse (b) to the reading in which Captain Spaulding did the shooting in his pajamas.

Why is NLP hard?

Natural languages are highly ambiguous at all levels

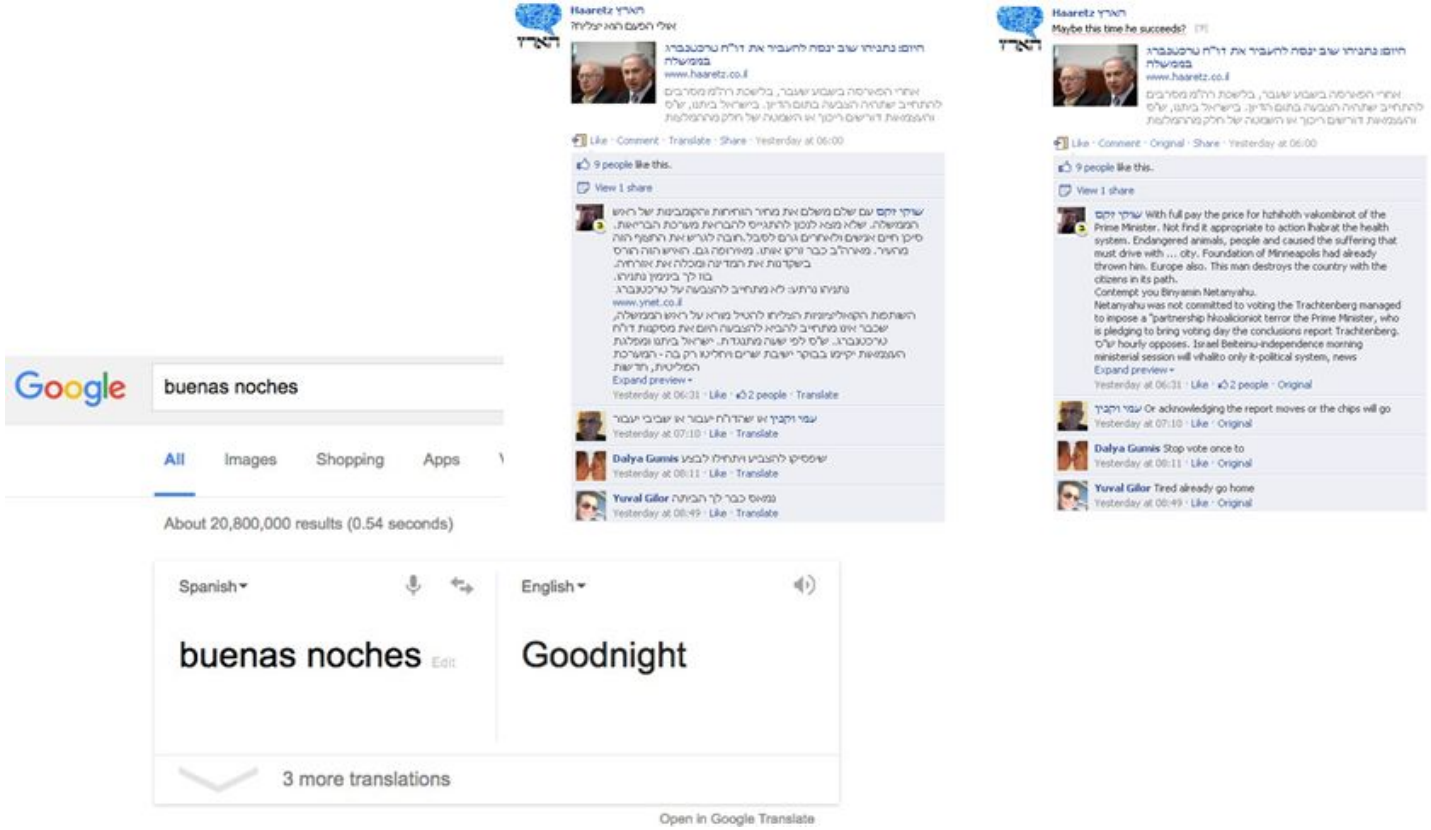
- Lexical (word's meaning)
- Syntactic
- Semantic
- Discourse

- Natural languages are fuzzy
- Natural languages involve reasoning about the world..E.g., It is unlikely that an elephant wears a pajamas

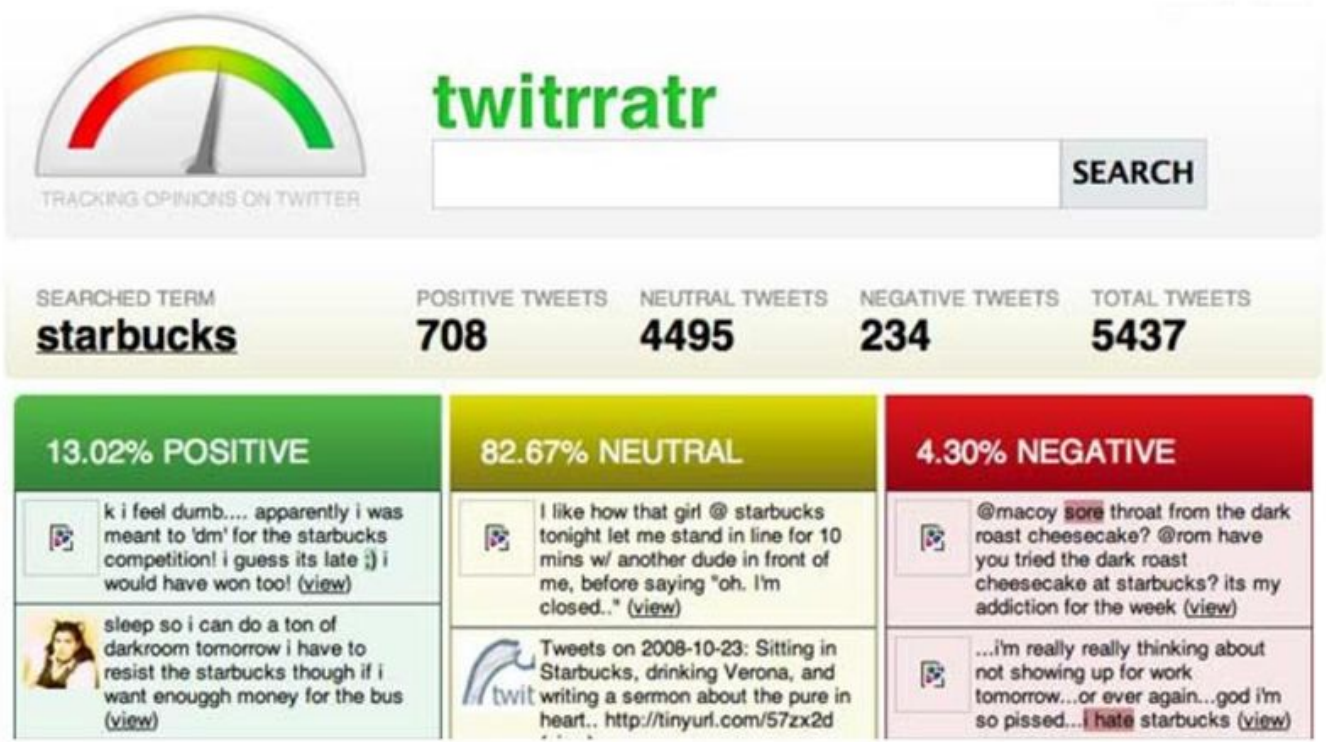
Applications of NLP

Machine

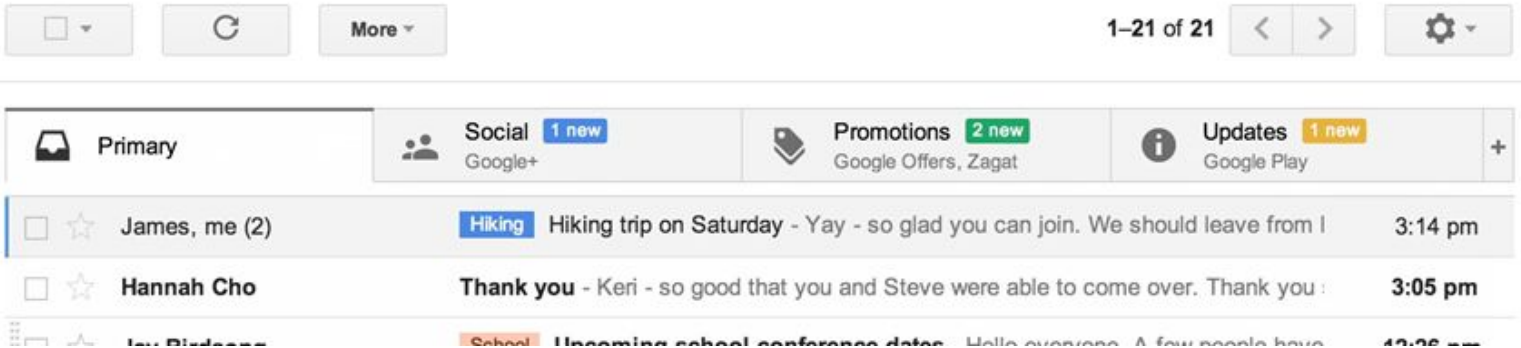
Translation



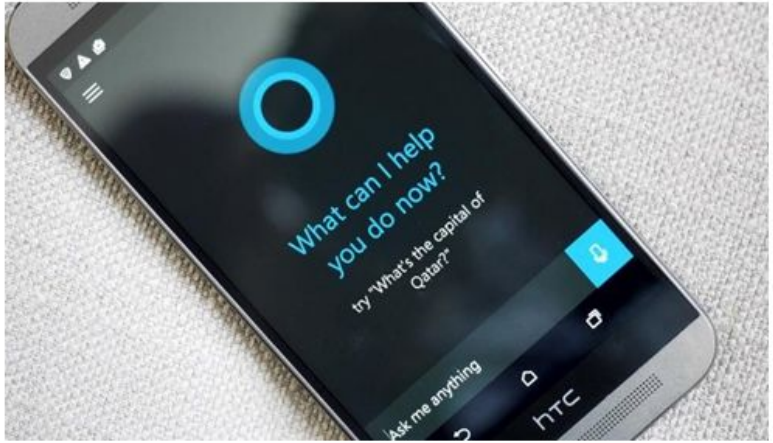
Sentiment/Opinion Analysis



Text Classification



Digital Personal Assistant



- Semantic parsing – understand tasks
- Entity linking – “my wife” = “Kellie” in the phone book

Brief history of NLP

Foundational Insights: 1940s and 1950s

− Two foundational paradigms

● Automaton

● Probabilistic/Information-Theoretic models

● The two camps: 1957-1970

− Symbolic paradigm: the work of Chomsky and others on

formal language theory and generative syntax (1950s ~mid 1960s)

- − Stochastic paradigm In departments of statistics

Brief history of NLP

Four paradigms: 1970-1983, explosion in

research in speech and language processing

- Stochastic paradigm
- Logic-based paradigm
- Natural language understanding
- Discourse modeling paradigm

● Empiricism and Finite State Models Redux: 1983-1993

Brief history of NLP

The Fields Comes Together: 1994-1999

− Probabilistic and data-driven models had become

quite standard

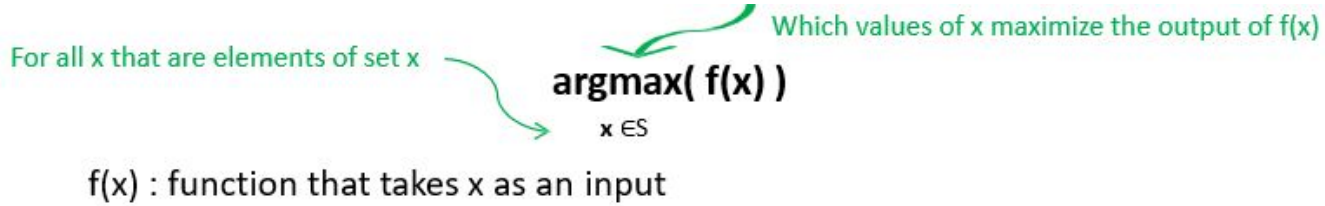
● The Rise of Machine Learning: 2000-now

− Large amount of spoken and textual data become available

− Widespread availability of high-performance computing systems

Argmax function

- The argmax() equation returns the largest values that maximize a function.



- Condition is met when the output of the function for at least one element xj that is greater than or equal to the output of for all x ∈ S.

The output for x_j is greater than or equal to the output for all elements x_i in set S

$$f(x_j) \geq f(x_i),\ x \in S$$

Bayesian Decision Theory

Bayes Theorem : Given the random variables A and B,

$$P(A \mid B) = \frac{P(A)P(B \mid A)}{P(B)}$$

$P(A \mid B)$ Posterior probability

$P(A)$ Prior probability

$P(B \mid A)$ Likelihood

Speech Recognition as a Noisy Channel

Speaker has word sequence W

W is articulated as acoustic sequence A

• This process introduces noise:

– variation in pronunciation

– acoustic variation due to microphone etc.

• Bayes theorem gives us:

$$\begin{aligned}\overline{W} &= \arg\max_W P(W|A) \\ &= \arg\max_W \underbrace{P(A|W)}_{likelihood}\underbrace{P(W)}_{prior}\end{aligned}$$

Collocations

- Collocation or lexical collocation means two or more words co-occur in a sentence more frequently than by chance. A collocation is an expression that forms a specific meaning. It may be noun phrase like large house, verbal phrase like pick up, idioms, cliches or technical terms.
- For example, He is known for his fair and square dealings and everybody trusts his work.
- Here fair and square means honest but if we take the individual words though the word fair gives somewhat closer meaning as it means just the word square confuses us. So instead of taking individual words one should take the collocation fair and square and find meaning. It shows that collocations play a key role in understanding sentences.

Collocations - contd

- It is a phrase consisting of more than one word but these words more commonly co-occur in a given context than its individual word parts. For example, in a set of hospital related documents, the phrase ‘CT scan’ is more likely to co-occur than do ‘CT’ and ‘scan’ individually. ‘CT scan’ is also a meaningful phrase.
- The two most common types of collocation are bigrams and trigrams. Bigrams are two adjacent words, such as ‘CT scan’, ‘machine learning’, or ‘social media’. Trigrams are three adjacent words, such as ‘out of business’, or ‘Proctor and Gamble’.

Uses of Collocation

a) Keyword extraction: identifying the most relevant keywords in documents to assess what aspects are most talked about

b) Bigrams/Trigrams can be concatenated (e.g. social media -> social_media) and counted as one word to improve insights analysis, topic modeling, and create more meaningful features for predictive models in NLP problems

Principal Approaches to Finding Collocations

- Selection of collocations by frequency
- PMI(Pointwise Mutual Information)
- Hypothesis testing (t-test and chi square)

Counting frequencies of adjacent words with part of speech filters

- The simplest method is to rank the most frequent bigrams or trigrams:

```
bigram_freq = bigramFinder.ngram_fd.items()

bigramFreqTable = pd.DataFrame(list(bigram_freq),
columns=['bigram','freq']).sort_values(by='freq', ascending=False)
```

However, a common issue with this is adjacent spaces, stop words, articles, prepositions or pronouns are common and are not meaningful:

	bigram	freq
0	(, -pron-)	29078
1	(, the)	21918
2	(-pron-, be)	18353
3	(the, room)	8898
4	(-pron-, have)	8377

	trigram	freq
0	(, -pron-, be)	6267
1	(the, room, be)	4411
2	(, the, room)	3349
3	(, -pron-, have)	2681
4	(the, staff, be)	2641

To fix this, we filter out for collocations not containing stop words and filter for only the following structures:

Bigrams: (Noun, Noun), (Adjective, Noun)

Trigrams: (Adjective/Noun, Anything, Adjective/Noun)

Pointwise Mutual Information (PMI)

The Pointwise Mutual Information (PMI) score for bigrams is:

$$PMI(w^1, w^2) = log_2 \frac{P(w^1, w^2)}{P(w^1)P(w^2)}$$

For trigrams:

$$PMI(w^1, w^2, w^3) = log_2 \frac{P(w^1, w^2, w^3)}{P(w^1)P(w^2)P(w^3)}$$

PMI

The main intuition is that it measures how much more likely the words co-occur than if they were independent. However, it is very sensitive to rare combination of words. For example, if a random bigram ‘abc xyz’ appears, and neither ‘abc’ nor ‘xyz’ appeared anywhere else in the text, ‘abc xyz’ will be identified as highly significant bigram when it could just be a random misspelling or a phrase too rare to generalize as a bigram. Therefore, this method is often used with a frequency filter.

For further reference on collocations:

https://medium.com/@nicharuch/collocations-identifying-phrases-that-act-like-individual-words-in-nlp-f58a93a2f84a

Terminology

Corpus:

In linguistics and NLP, corpus (literally Latin for body) refers to a collection of texts.

Lexicon:

The definition of a lexicon is a dictionary or the vocabulary of a language, a people or a subject. An example of lexicon is YourDictionary.com. An example of lexicon is a set of medical terms.

Bag of Words

- Bag of Words (BOW) is a method to extract features from text documents. These features can be used for training machine learning algorithms. It creates a vocabulary of all the unique words occurring in all the documents in the training set.

1. "John likes to watch movies. Mary likes movies too."

2. "John also likes to watch football games."

1. ['John', 'likes', 'to', 'watch', 'movies.', 'Mary', 'likes', 'movies', 'too.']

2. ['John', 'also', 'likes', 'to', 'watch', 'football', 'games']

1. {"John":1,"likes":2,"to":1,"watch":1,"movies":2,"Mary":1,"too":1}

2. {"John":1,"also":1,"likes":1,"to":1,"watch":1,"football":1, "games":1}

{"John":2,"likes":3,"to":2,"watch":2,"movies":2,"Mary":1,"too":1,
"also":1,"football":1,"games":1}

The length of the vector will always be equal to vocabulary size. In this case the vector length is 11.In order to represent our original sentences in a vector, each vector is initialized with all zeros — [0, 0, 0, 0, 0, 0, 0, 0, 0, 0]

John likes to watch movies. Mary likes movies too.[1, 2, 1, 1, 2, 1, 1, 0, 0, 0]

John also likes to watch football games.[1, 1, 1, 1, 0, 0, 0, 1, 1, 1]

Limitations of BOW

Semantic meaning: the basic BOW approach does not consider the meaning of the word in the document. It completely ignores the context in which it's used. The same word can be used in multiple places based on the context or nearby words.

Vector size: For a large document, the vector size can be huge resulting in a lot of computation and time. You may need to ignore words based on relevance to your use case.

TF-IDF(Term Frequency-Inverse Document Frequency)

- Tf-idf stands for term frequency-inverse document frequency, and the tf-idf weight is a weight often used in information retrieval and text mining.
- This weight is a statistical measure used to evaluate how important a word is to a document in a collection or corpus.
- The importance increases proportionally to the number of times a word appears in the document but is offset by the frequency of the word in the corpus.
- Variations of the tf-idf weighting scheme are often used by search engines as a central tool in scoring and ranking a document's relevance given a user query.

TF

Term Frequency, which measures how frequently a term occurs in a document. Since every document is different in length, it is possible that a term would appear much more times in long documents than shorter ones. Thus, the term frequency is often divided by the document length (aka. the total number of terms in the document) as a way of normalization:

TF(t) = (Number of times term t appears in a document) / (Total number of terms in the document).

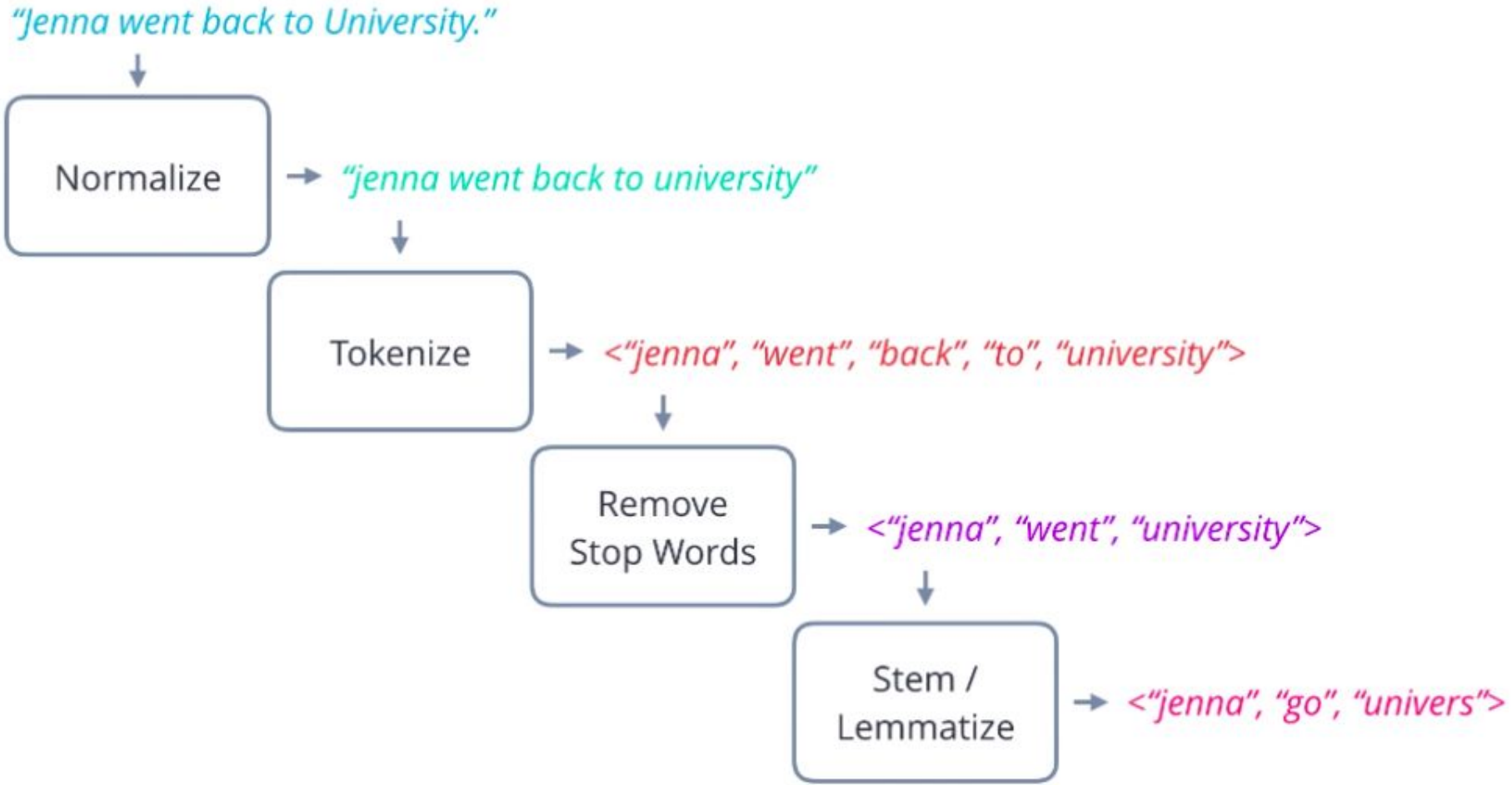
IDF

Inverse Document Frequency, which measures how important a term is. While computing TF, all terms are considered equally important. However it is known that certain terms, such as "is", "of", and "that", may appear a lot of times but have little importance. Thus we need to weigh down the frequent terms while scale up the rare ones, by computing the following:

IDF(t) = log_e(Total number of documents / Number of documents with term t in it).

Example

Consider a document containing 100 words wherein the word cat appears 3 times. The term frequency (i.e., tf) for cat is then (3 / 100) = 0.03. Now, assume we have 10 million documents and the word cat appears in one thousand of these. Then, the inverse document frequency (i.e., idf) is calculated as log(10,000,000 / 1,000) = 4. Thus, the Tf-idf weight is the product of these quantities: 0.03 * 4 = 0.12.


"Jenna went back to University."
Normalize
"jenna went back to university"
Tokenize
<"jenna", "went", "back", "to", "university">
Remove Stop Words
<"jenna", "went", "university">
Stem / Lemmatize
<"jenna", "go", "univers">"

Tokenization

- *Tokenization is breaking a text chunk in smaller parts. Whether it is breaking Paragraph in sentences, sentence into words or word in characters.*
- *https://medium.com/data-science-in-your-pocket/tokenization-algorithms-in-natural-language-processing-nlp-1fceab8454af*
- *https://analyticsindiamag.com/hands-on-guide-to-different-tokenization-methods-in-nlp/*
-

Stemming

Stemming is the process of producing morphological variants of a root/base word. Stemming programs are commonly referred to as stemming algorithms or stemmers.

A stemming algorithm reduces the words “chocolates”, “chocolatey”, “choco” to the root word, “chocolate” and “retrieval”, “retrieved”, “retrieves” reduce to the stem “retrieve”. Stemming is an important part of the pipelining process in Natural language processing. The input to the stemmer is tokenized words.

https://www.geeksforgeeks.org/introduction-to-stemming/

Lemmatization

Lemmatization is the process of grouping together the different inflected forms of a word so they can be analysed as a single item. Lemmatization is similar to stemming but it brings context to the words. So it links words with similar meaning to one word.

Examples of lemmatization:

-> rocks : rock,corpora : corpus,better : good

Diff b/w stemming and lemmatization

https://towardsdatascience.com/lemmatization-in-natural-language-processing-nlp-and-machine-learning-a4416f69a7b6