NLP Intro

Week 08 - Day 01

What is NLP

Extracting features and insights from unstructured text

Enabling computers to understand language the same way people do

Is NLP really used?

Google (information retrieval)

Siri (speech recognition)

Twitter & Reviews (sentiment analysis)

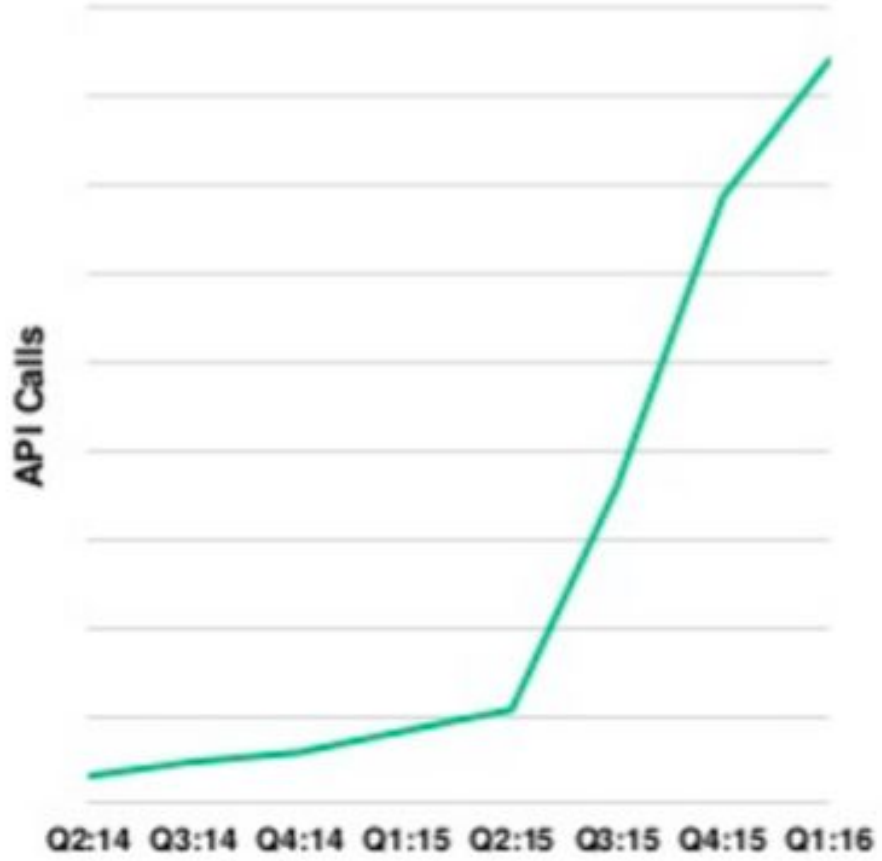
Google Translate (machine translation)

Autocorrect

*As speech recognition accuracy goes from say 95% to 99%, all of us in the room will go from barely using it today to using it all the time. Most people underestimate the difference between 95% and 99% accuracy – **99% is a game changer**...*

Andrew Ng

Baidu Text to Speech (TTS) Daily Usage by API Calls, Global, 2014 – 2016[2]



NLP is Hard!

"High School Dropouts Cut in Half”

Brandy Jensen
@BrandyLJensen

I appreciate that zuck has eight gazillion dollars and still looks like he got his hair cut by his mom

https://twitter.com/ashleyfeinberg/status/983780105792245761?ref_src=twsrc%5Etfw&ref_url=https%3A%2F%2Fhellogiggles.com%2Fnews%2Fmark-zuckerberg-hearing-congress-facebook-memes%2F

Spam or Ham?

Hello, I saw your contact information on LinkedIn. I have carefully read through your profile and you seem to have an outstanding personality. This is one major reason why I am in contact with you. My name is Mr. Valery Grayfer Chairman of the Board of Directors of PJSC "LUKOIL". I am 86 years old and I was diagnosed with cancer 2 years ago. I will be going in for an operation later this week. I decided to WILL/Donate the sum of 8,750,000.00 Euros (...)

Hello, I am writing in regards to your application to the position of Data Scientist at Hooli X. We are pleased to inform you that you passed the first round of interviews and we would like to invite you for an on-site interview with our Senior Data Scientist Mr. John Smith. You will find attached to this message further information on date, time and location of the interview. Please let me know if I can be of any further assistance. Best Regards.

Bag of Words

“You are just a fuc*ing bag of words!”

“Ciao, il mio nome e’ Edoardo. Ciao!”

(ciao=2, il=1, mio=1, nome=1, … ,edoardo=1)

Document 1 -> [0,0,1,0,0,2,2,...,1]

Document 2 -> [1,0,2,0,1,0,0,...,0]

...

Document n -> [.....]

Sklearn - CountVectorizer

tokenizer : callable or None (default)

Override the string tokenization step while preserving the preprocessing and n-grams generation steps.

stop_words : string {‘english’}, list, or None (default)

If ‘english’, a built-in stop word list for English is used.

If a list, that list is assumed to contain stop words, all of which will be removed from the resulting tokens.

max_df : float in range [0.0, 1.0] or int, default=1.0

When building the vocabulary ignore terms that have a document frequency strictly higher than the given threshold (corpus-specific stop words). If float, the parameter represents a proportion of documents, integer absolute counts.

min_df : float in range [0.0, 1.0] or int, default=1

When building the vocabulary ignore terms that have a document frequency strictly lower than the given threshold. This value is also called cut-off in the literature. If float, the parameter represents a proportion of documents, integer absolute counts. This parameter is ignored if vocabulary is not None.

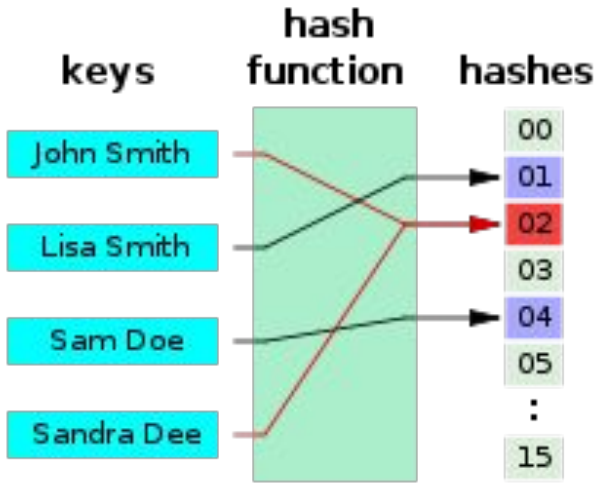
ngram_range : tuple (min_n, max_n)

The lower and upper boundary of the range of n-values for different n-grams to be extracted. All values of n such that min_n <= n <= max_n will be used.

“Ngrams are contiguous sequences of n items.”

“Ngrams are”, “are contiguous”, “contiguous sequences”, “sequences of”, “of n”, “n items”

Alternative = HashingVectorizer



TF-IDF

Term Frequency

+

Inverse Document Frequency

“Alternative” to simple term frequency

Importance of a word

in a document in a corpus

=

Importance(word,document)

+

Importance(word,corpus)

Example: Search Engine

Document 1 -> 5% of words are “dog”

Document 2 -> 0.001% of the words are “dog”

Query “dog” -> more relevant document?

Document 1 -> 5% of words are “dog”

Document 2 -> 0.001% of the words are “dog”

Query “dog” -> more relevant document?

Document 1 -> 5% of words are “dog”

Document 2 -> 0.001% of the words are “dog”

Term frequency!

If all the documents contains the word “like”, is “like” a useful feature?

Inverse Document frequency!

*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.

We have 10M documents and the word ***cat*** *appears in 1000 of these.*

The inverse document frequency (i.e., idf) is calculated as :

log(10,000,000 / 1,000) = 4.

Less frequent = More important

$$\log(10{,}000{,}000 / 1{,}000) = 4$$

$$\log(10{,}000{,}000 / 10{,}000) = 3$$

More frequent = Less important

Tf-idf weight = tf * idf = 0.03 * 4 = 0.12

Tf-idf weight = tf * idf = 0.03 * 4 = 0.12

Stemming

Swimmed -> Swim

Swimming -> Swim

Am, are, is -> Be

Car, cars, car’s, cars’ -> Car

“the boy's cars are different colors” ->

the boy car be differ color

Part-of-Speech Tagging

“The” -> article

“Good” -> adjective

“Guitar” -> noun

Pipeline Example

Tokenization -> Remove stop-words -> Stemming -> Add bigrams -> TF-IDF -> add PoS -> classification

Libraries

Sklearn

NLTK

..etc..

Summary

- NLP = techniques to understand text
- NLP is hard!
- Bag of words
- TF-IDF
- Stop-words, Token, Stemming, N-Grams, PoS tagging