GENERAL ASSEMBLY

More NLP - Feature Extraction from Text

Patrick Smith

More NLP - Feature Extraction from Text

LEARNING OBJECTIVES

- Extract features from free form text using Scikit Learn
- Identify Parts of Speech using NLTK
- Remove stop words
- Describe how TFIDF works

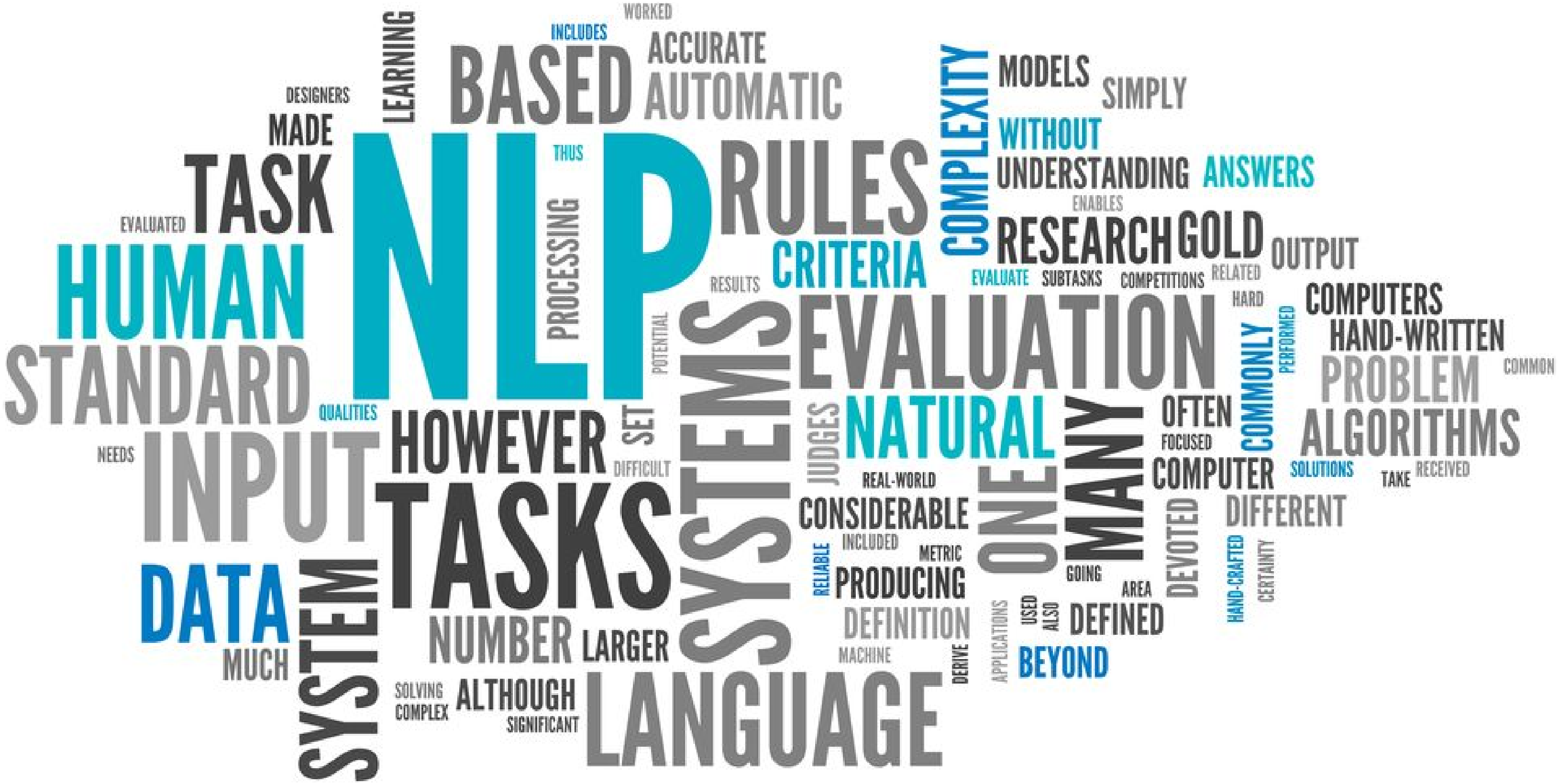
More NLP - Feature Extraction from Text

PRE-WORK

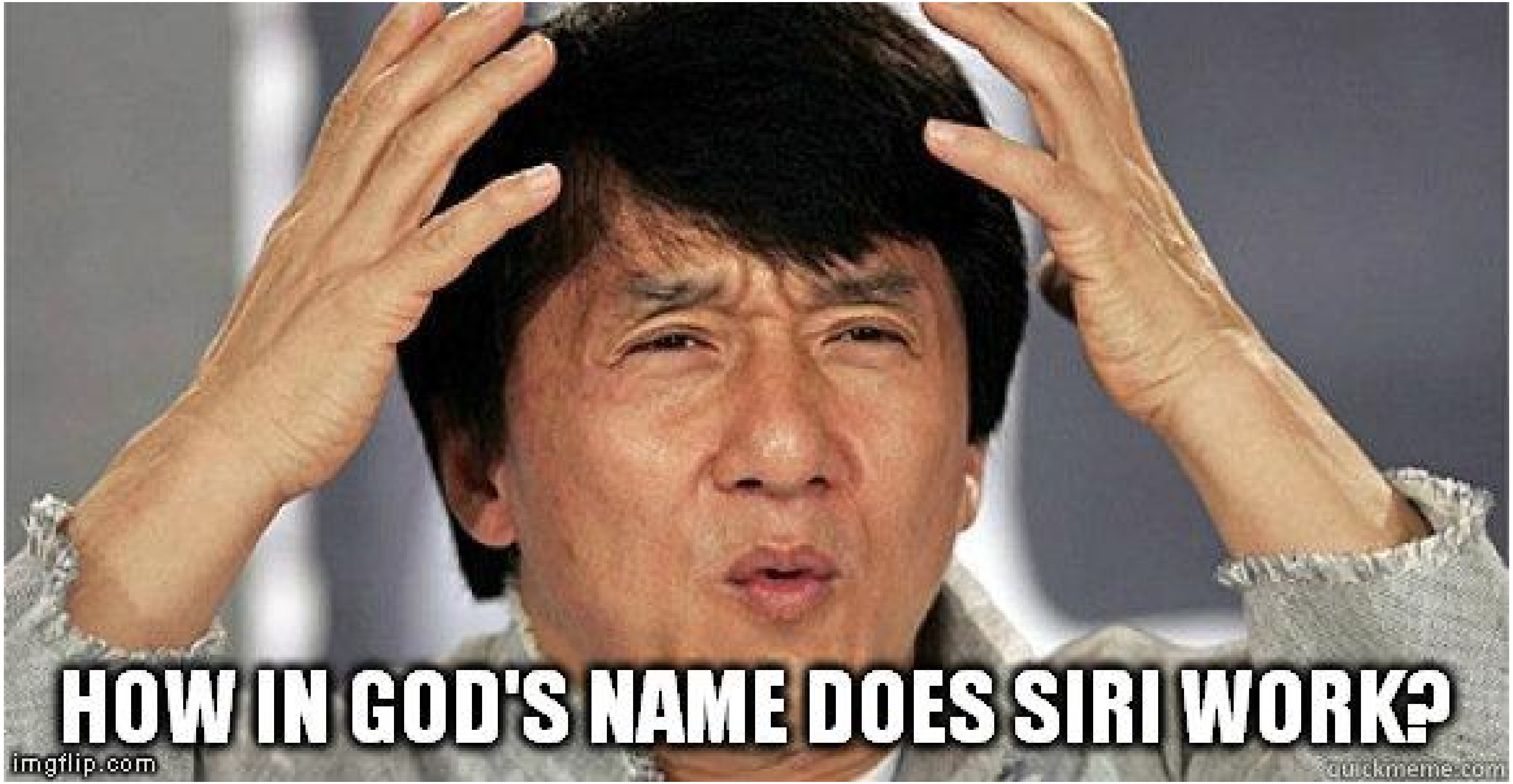
- Familiarize yourself with nltk.download() in case you need to download additional corpuses
- Describe what a transformer is in Scikit Learn and use it
- Recognize basic principles of English language syntax

OPENING

More NLP - Feature Extraction from Text



More NLP - Feature Extraction from Text



More NLP - Feature Extraction from Text

All the models we have learned so far accept a 2D table of real numbers as input (we called it X) and output a vector of classes or numbers (we called it y).

More NLP - Feature Extraction from Text

All the models we have learned so far accept a 2D table of real numbers as input (we called it X) and output a vector of classes or numbers (we called it y).

Very often though, our starting point data is not given in the form of a table of numbers, rather it's unstructured, for example in the case of text documents.

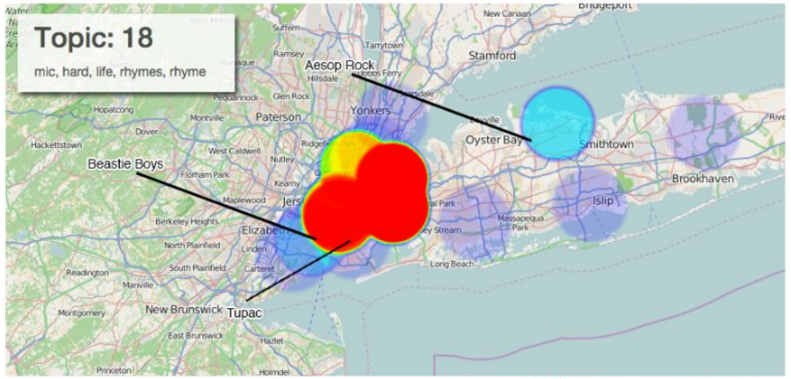
More NLP - Feature Extraction from Text

In this case we need a way to go from unstructure data to a table of numbers, so that we can then apply the usual methods. This is called feature extraction and this lesson is dedicated to it.

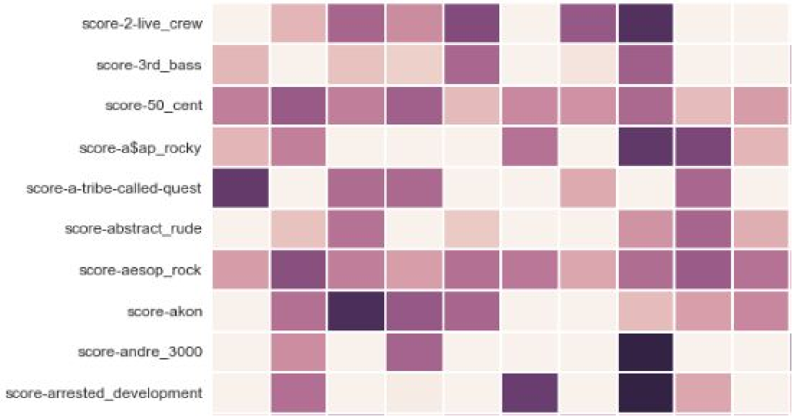
More NLP - Feature Extraction from Text

Case in Point: Rapstats

HIP HOP
AN ANALYSIS OF LYRICS USING
MACHINE LEARNING TECHNIQUES /
VOL.1
PARENTAL ADVISORY EXPLICIT CONTENT

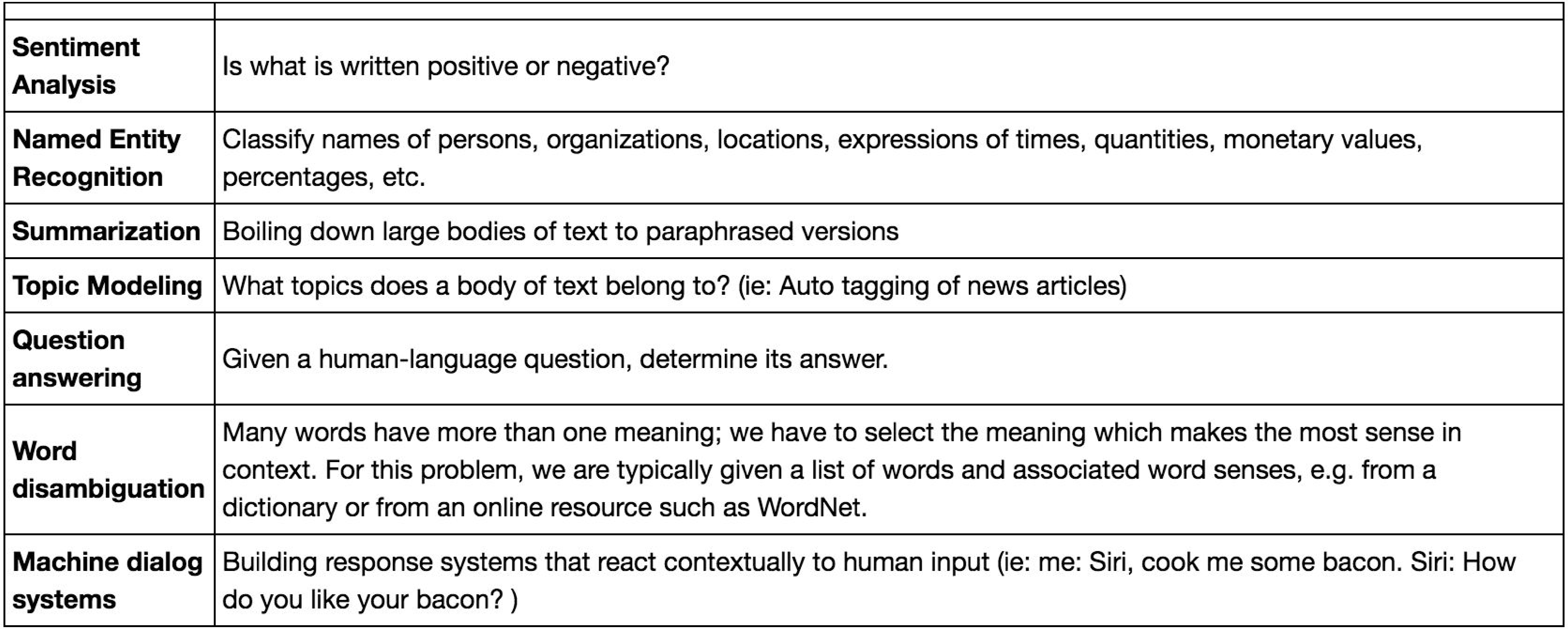


Red = High Probability of Artist in Topic



Review: Feature Extraction from text

More NLP - Feature Extraction from Text



Sentiment Analysis	Is what is written positive or negative?
Named Entity Recognition	Classify names of persons, organizations, locations, expressions of times, quantities, monetary values, percentages, etc.
Summarization	Boiling down large bodies of text to paraphrased versions
Topic Modeling	What topics does a body of text belong to? (ie: Auto tagging of news articles)
Question answering	Given a human-language question, determine its answer.
Word disambiguation	Many words have more than one meaning; we have to select the meaning which makes the most sense in context. For this problem, we are typically given a list of words and associated word senses, e.g. from a dictionary or from an online resource such as WordNet.
Machine dialog systems	Building response systems that react contextually to human input (ie: me: Siri, cook me some bacon. Siri: How do you like your bacon?)

More NLP - Feature Extraction from Text

Email Example

More NLP - Feature Extraction from Text

Check: Can you think of a simple heuristic rule to catch email like this?

More NLP - Feature Extraction from Text

Bag of words / Word Counting

The bag-of-words model is a simplifying representation used in natural language processing. In this model, a text (such as a sentence or a document) is represented as the bag (multiset) of its words, disregarding grammar and even word order but keeping multiplicity.

What happens when we get a character that is referenced outside of the character space, for instance a Japanese Katakana character? (片仮名 / カタカナ)¶

More NLP - Feature Extraction from Text

- Python doesn't know how to handle these characters if it has to process it in any way
- Characters outside the latin character space will get converted to ordinal 0
- This problem can be very frustrating to deal with
- Use sklearns built-in text preprocessing method when possible.
- Always save/encode your text as UTF8 when there are options available to do so.

Review: Scikit Learn Count Vectorizer

More NLP - Feature Extraction from Text

Intro: Scikit-Learn Hashing Vectorizer

More NLP - Feature Extraction from Text

Scikit-Learn Hashing Vectorizer

A hash value (or simply hash), also called a message digest, is a number generated from a string of text. The hash is substantially smaller than the text itself, and is generated by a formula in such a way that it is extremely unlikely that some other text will produce the same hash value.

Scikit-Learn Hashing Vectorizer

As you have seen we can set the CountVectorizer dictionary to have a fixed size, only keeping words of certain frequencies

- However, we still have to compute a dictionary and hold the dictionary in memory.
- This could be a problem when we have a large corpus or in streaming applications where we don't know which words we will encounter in the future.

More NLP - Feature Extraction from Text

Scikit-Learn Hashing Vectorizer

These problems can be solved using the HashingVectorizer, which converts a collection of text documents to a matrix of occurrences, calculated with the hashing trick.

- Each word is mapped to a feature with the use of a hash function that converts it to a hash.
- If we encounter that word again in the text, it will be converted to the same hash, allowing us to count word occurrence without retaining a dictionary in memory.

More NLP - Feature Extraction from Text

What characteristics should feature extraction from text satisfy?

More NLP - Feature Extraction from Text

Using the code from before, let's repeat the vectorization using a HashingVectorizer.

More NLP - Feature Extraction from Text

Intro: Natural Language Processing

Natural Language Processing

Bag of word approaches like the one outlined before completely ignore the structure of a sentence, they merely assess presence of specific words or word combinations.

Natural Language Processing

Bag of word approaches like the one outlined before completely ignore the structure of a sentence, they merely assess presence of specific words or word combinations.

Besides, the same word can have multiple meanings in different contexts. Consider for example the following two sentences:

- There's wood floating in the sea
- Mike's in a sea of trouble with the move

Segmentation

Segmentation is a technique to identify sentences within a body of text. Language is not a continuous uninterrupted stream of words: punctuation serves as a guide to group together words that convey meaning when contiguous.

More NLP - Feature Extraction from Text

Does NLTK offer other Tokenizers? Use *nltk.download()* to explore the available packages.

More NLP - Feature Extraction from Text

Advanced NLP with NLTK

Advanced NLP

In the case of NLP, normalization is sometime required when slightly different version of a word exist. For example: LinkedIn sees 6000+ variations of the title "Software Engineer" and 8000+ variations of the word "IBM".

Check: What are other common cases of text that could need normalization?

More NLP - Feature Extraction from Text

Advanced NLP

It would be wrong to consider the words "MR." and "mr" to be different features, thus we need a technique to normalize words to a common root.

This technique is called Stemming.

Science, Scientist => Scien
Swimming, Swimmer, Swim => Swim

More NLP - Feature Extraction from Text

Group Activity (10 Min)

More NLP - Feature Extraction from Text

Term frequency - Inverse document Frequency (10 mins-ish)

Term Frequency

More interesting than stop-words is the tf-idf score. This tells us which words are most discriminating between documents. Words that occur a lot in one document but doesn't occur in many documents will tell you something special about the document.

Term Frequency

- This weight is a statistical measure used to evaluate how important a word is to a document in a collection (aka 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.

More NLP - Feature Extraction from Text

Term Frequency

Let's see how it is calculated.

Term frequency tf is the frequency of a certain term in a document:

$$\mathrm{tf}(t, d) = \frac{N_{\text{term}}}{N_{\text{terms in Document}}} \Bigg|$$

Inverse document frequency is defined as the frequency of documents that contain that term over the whole corpus.

$$\mathrm{idf}(t, D) = \log$$

$$\frac{N_{\text{Documents}}}{N_{\text{Documents that contain term}}}$$

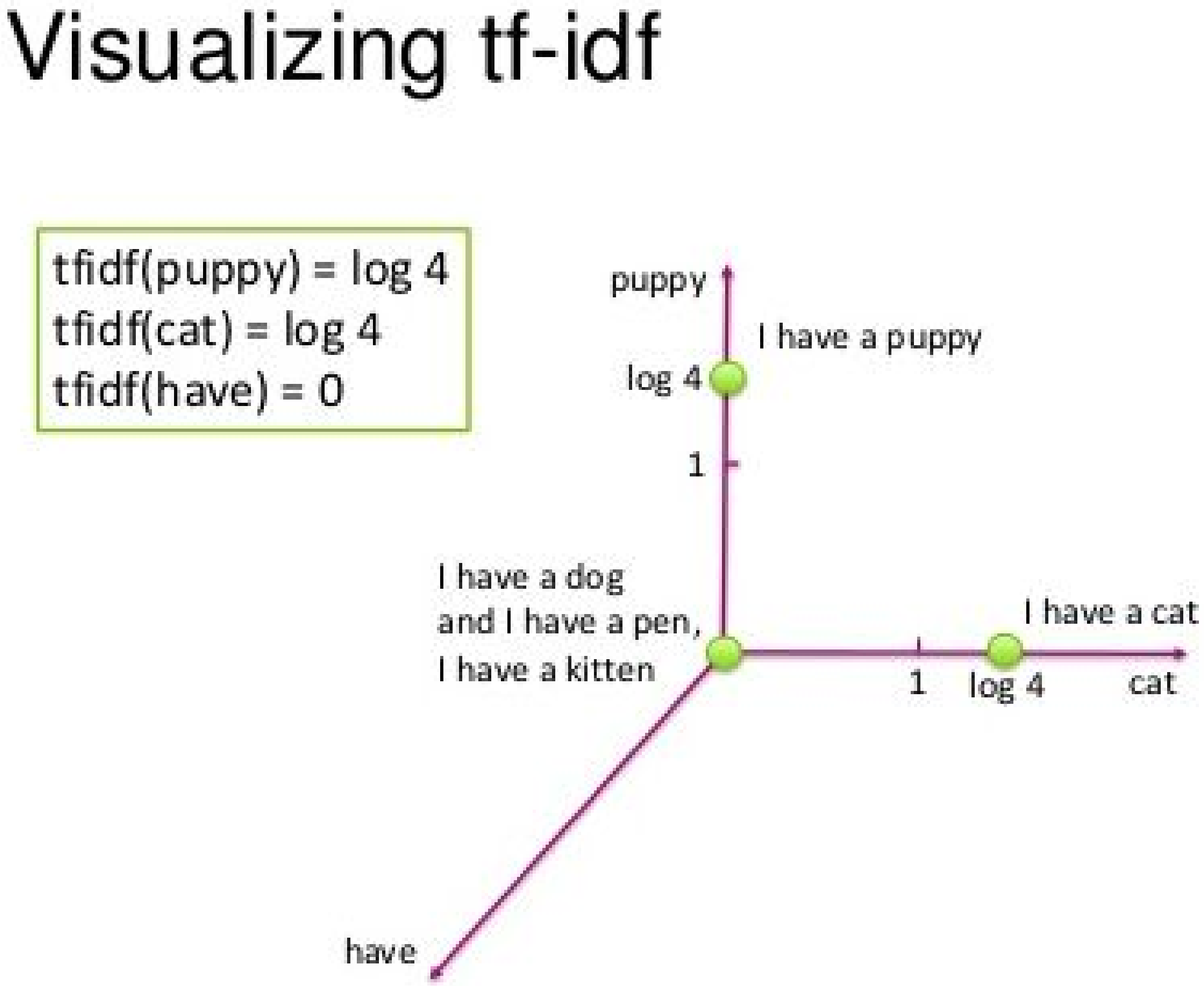
Term frequency - Inverse Document Frequency is calculated as:

$$\mathrm{tfidf}(t, d, D) = \mathrm{tf}(t, d)$$

$$\cdot\ \mathrm{idf}(t, D)$$

More NLP - Feature Extraction from Text

Term Frequency



Visualizing tf-idf

tfidf(puppy) = log 4
tfidf(cat) = log 4
tfidf(have) = 0

Term Frequency

This enhances terms that are highly specific of a particular document, while suppressing terms that are common to most documents.

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.

Term Frequency

The inverse document frequency is a measure of how much information the word provides, that is, whether the term is common or rare across all documents.

It is the logarithmically scaled inverse fraction of the documents that contain the word, obtained by dividing the total number of documents by the number of documents containing the term, and then taking the logarithm of that quotient

More NLP - Feature Extraction from Text

Can you think of situations where the definition above may be problematic?

More NLP - Feature Extraction from Text

Practice

Conclusion

More NLP - Feature Extraction from Text

Q & A

More NLP - Feature Extraction from Text

Review