BASIC TEXT REPRESENTATION

BY

Wuraola Oyewusi

What is Text Data?

Text data usually consists of documents which can represent words, sentences or even paragraphs of free flowing text. The inherent unstructured (no neatly formatted data columns!) and noisy nature of textual data makes it harder for machine learning methods to directly work on raw text data.

Dipanjan (DJ) Sarkar, January 2019

Some of the most important and interesting data that you will encounter in practice will come in text form.

For Examples,

Sentiment analysis of social media text, Topic modelling , Basic Text Classification, e.t.c

How does computer perceive text?

Raw Text to Feature Representation to ML Algorithm

How does computer perceive text?

The classifiers and learning algorithms can not directly process the text documents in their

original form, as most of them expect numerical feature vectors with a fixed size rather than

the raw text documents with variable length.

The process of transforming text into numeric stuff, is usually performed by building a language model. These models typically assign probabilities, frequencies or some obscure numbers to words, sequences of words, group of words, section of documents or whole documents

Michel Kana, Ph.D, July 15,2018

Techniques of Basic Text Representation

1-hot - encoding Model

N-Grams Model

Bag-of-words Model

TF-IDF Model

1-hot encoding Model

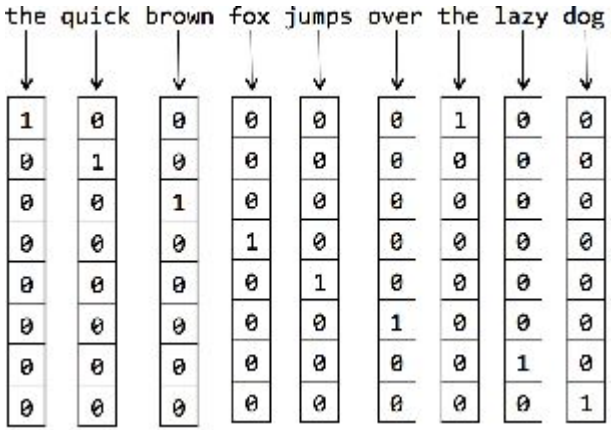
A one hot encoding is a representation of categorical variables/tokens as binary vectors.

This first requires that the categorical values be mapped to integer values.

Then, each integer value is represented as a binary vector that is all zero values except the index of the integer, which is marked with a 1.

1-hot encoding Model

source: Fundamentals of Deep Learning, N. Buduma, 2017



1-hot encoding Model

Pros

Simple and intuitive.

Cons

Ineffective for Large Vocabularies

Misses the relationship between words

N-grams Model

N-gram language models estimate the probability of the last words given the previous words. It finds use in spell checking,auto completion,language identification,text generation etc.

Sentence = “Welcome to AISaturday Lagos”

1-gram(or unigram) : “Welcome”, “to”, “AISaturday”, “Lagos”

2-gram (or bigram) : “Welcome to”, “to AISaturday”, “AISaturday Lagos”

3-gram(or trigram) : “Welcome to AISaturday”, “to AISaturday Lagos”

N-grams Model

The longer the context on which we train a N-gram model, the more coherent the sentences we can generate

Furthermore, the N-gram model is heavily dependent on the training corpus used to calculate the probabilities. One implication of this is that the probabilities often encode specific facts about a given training text, which may not necessarily apply to a new text

Bag of words Model

For tasks that are not based on sequential pattern of words maybe like classifying texts based on sentiments or detecting the language a text is written in. Texts can be represented by bag of words, ignoring their original position in the text, keep only their frequency.

This method relies on term frequency, the number of times a token shows up in a document is counted and this value is used as it’s weight

TF-IDF Model

TF-IDF stands for “term frequency-inverse document frequency”, meaning the weight assigned to each token not only depends on its frequency in a document but also how recurrent that term is in the entire corpora.

Term Frequency :

Term frequency = (Number of Occurrences of a word)/(Total words in the document)

Inverse document frequency:

IDF(word) = Log((Total number of documents)/(Number of documents containing the word))

Some Text Preprocessing Methods

- HTML Tag Removal
- Removal of accented and special characters
- Contraction Expansion
- Stemming and Lemmatization
- Stopwords Removal

DATASET

Drug Review Dataset (Druglib.com) Data Set

Attributes : 8 **Instances: 4143**

URL LINK : https://archive.ics.uci.edu/ml/datasets/Drug+Review+Dataset+%28Druglib.com%29

1. urlDrugName (categorical): name of drug
2. condition (categorical): name of condition
3. benefitsReview (text): patient on benefits
4. sideEffectsReview (text): patient on side effects
5. commentsReview (text): overall patient comment
6. rating (numerical): 10 star patient rating
7. sideEffects (categorical): 5 step side effect rating
8. effectiveness (categorical): 5 step effectiveness rating

References

https://towardsdatascience.com/understanding-feature-engineering-part-3-traditional-methods-for-text-data-f6f7d70acd41

https://archive.ics.uci.edu/ml/datasets/Drug+Review+Dataset+%28Druglib.com%29

https://towardsdatascience.com/representing-text-in-natural-language-processing-1eead30e57d8