ROLE OF CLASSIFIERS IN NLP

Alok Bhawankar PD09
Vivek Ray PD17
Anmol Singh PD26

Natural Language Processing

- Natural Language Processing (NLP) is a wide area of research where the worlds of artificial intelligence, computer science, and linguistics collide.

- It includes a bevy of interesting topics with cool real-world applications, like named entity recognition, machine translation or machine question answering.

- Each of these topics has its own way of dealing with textual data.

- Thus, classification plays an important step in Natural Language Processing.

Overview

What is Classifier?

How does it work?

How it affects NLP?

Classification

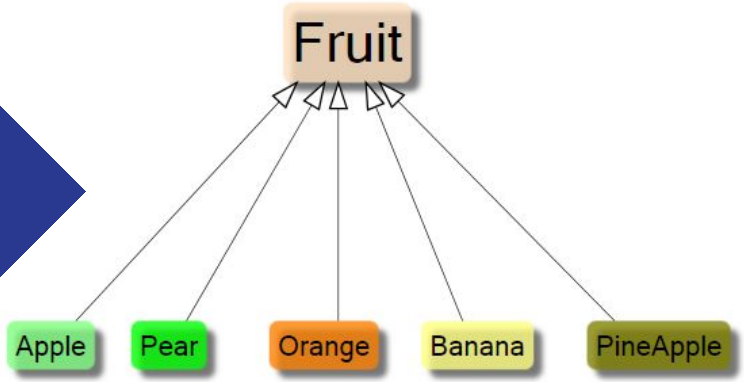
- A classifier is a hypothesis or discrete-valued function that is used to assign (categorical) class labels to particular data points.

- Classification is a supervised learning concept.

- The most common classification problems are speech recognition, face detection, handwriting recognition, document classification, etc.

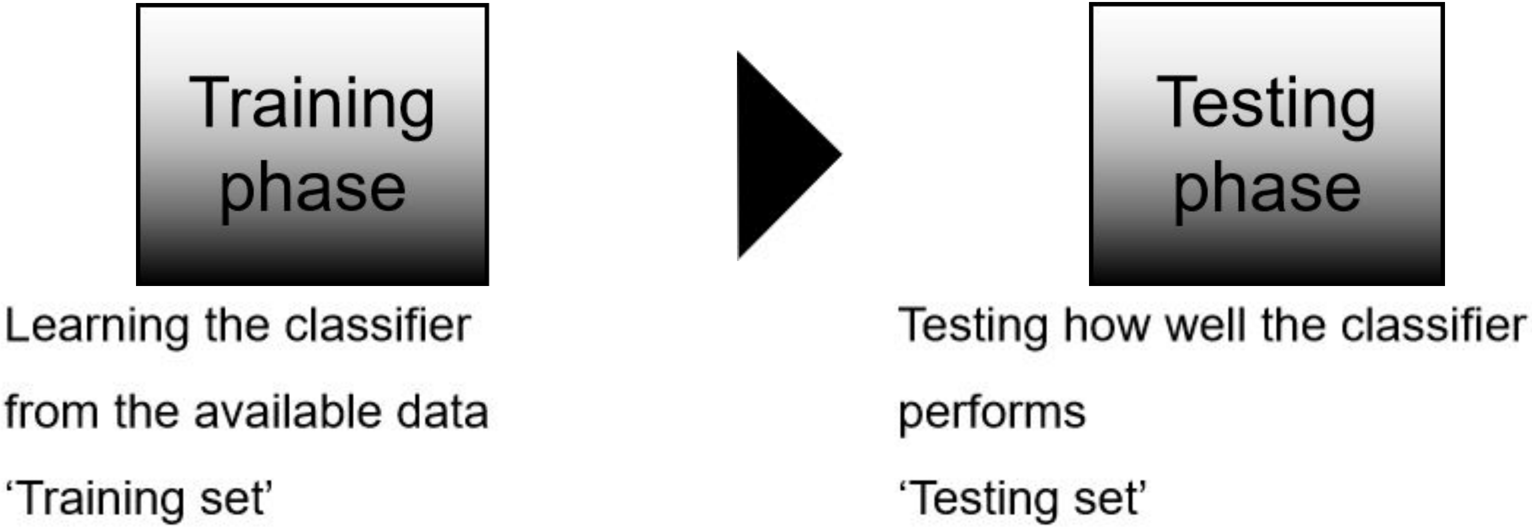
Example:

Classification



Working of classifier

- A classifier utilizes some training data to understand how given input variables relate to the class.
- When the classifier is trained accurately, it can be used to detect an unknown class.



Training phase

Testing phase

Learning the classifier from the available data

‘Training set’

Testing how well the classifier performs

‘Testing set’

Types Of Learners In Classification :-

Lazy learner:	**Eager learner:**
● Just store Data set without learning from it	● When it receive data set it starts classifying (learning)
● Start classifying data when it receive Test data	● Then it does not wait for test data to learn
● So it takes less time learning and more time classifying data	● So it takes long time learning and less time classifying data
● K - Nearest Neighbour, Case - Based Reasoning	● Decision Tree, Naive Bayes, Artificial Neural Networks

Types of Classification Algorithms

Linear Model
Logistic Regression Support Vector Machines

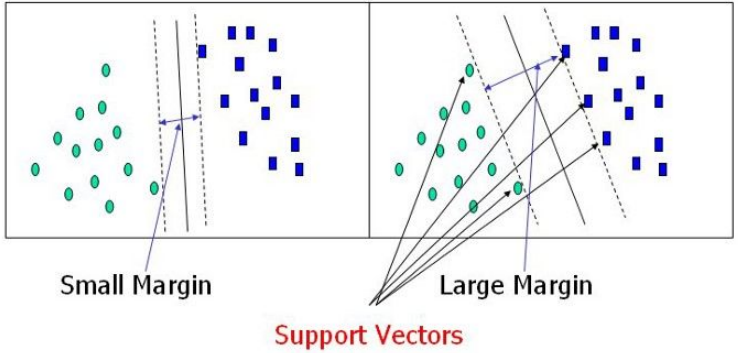
Nonlinear Model
K-nearest Neighbors (KNN) Kernel Support Vector Machines (SVM) Naïve Bayes Decision Tree Classification Random Forest Classification

Logistic Regression (Predictive Learning Model):

- It is a statistical method for analyzing a data set in which there are one or more independent variables that determine an outcome.
- The outcome is measured with a dichotomous variable (in which there are only two possible outcomes).
- The goal of logistic regression is to find the best fitting model to describe the relationship between the dichotomous characteristic of interest (dependent variable = response or outcome variable) and a set of independent (predictor or explanatory) variables.
- This is better than other binary classification like nearest neighbor since it also explains quantitatively the factors that lead to classification.

Support Vector Machine Algorithm

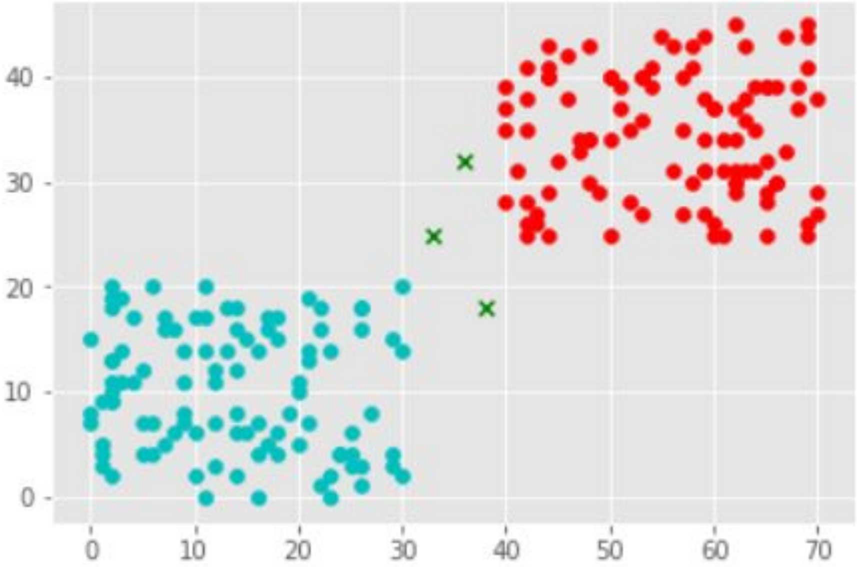
- The objective of the support vector machine algorithm is to find a hyperplane in an N-dimensional space(N — the number of features) that distinctly classifies the data points.



- To separate the two classes of data points, there are many possible hyperplanes that could be chosen.
- Our objective is to find a plane that has the maximum margin, i.e the maximum distance between data points of both classes. Maximizing the margin distance provides some reinforcement so that future data points can be classified with more confidence.

Nearest Neighbor :

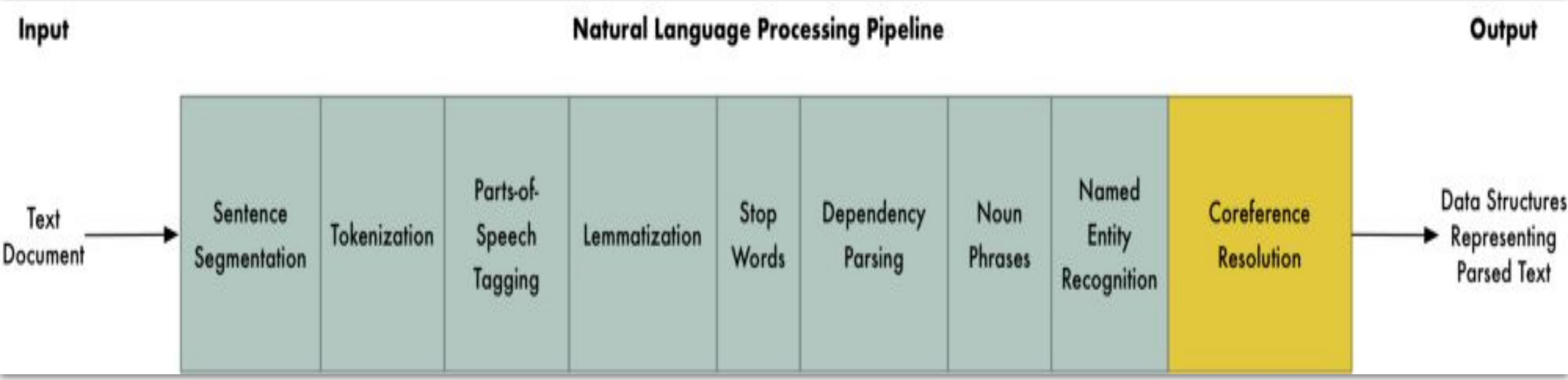
- The k-nearest-neighbors algorithm is a supervised classification technique that uses proximity as a proxy for ‘sameness’.
- The algorithm takes a bunch of labelled points and uses them to learn how to label other points.
- To label a new point, it looks at the labelled points closest to that new point (those are its nearest neighbors).
- Closeness is typically expressed in terms of a dissimilarity function.
- Once it checks with ‘k’ number of nearest neighbors, it assigns a label based on whichever label the most of the neighbors have.



Decision Trees:

- Decision tree builds classification or regression models in the form of a tree structure.
- It breaks down a data set into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed.
- The final result is a tree with decision nodes and leaf nodes. A decision node has two or more branches and a leaf node represents a classification or decision.
- The topmost decision node in a tree which corresponds to the best predictor called root node.
- Decision trees can handle both categorical and numerical data.

Natural Language Processing Pipeline



Role of classifiers in NLP

One of the most important sub-tasks in pattern classification are feature extraction and selection; the three main criteria of good features are listed below:

- Salient. The features are important and meaningful with respect to the problem domain.
- Invariant. Invariance is often described in context of image classification: The features are insusceptible to distortion, scaling, orientation, etc. A nice example is given by C. Yao and others in Rotation-Invariant Features for Multi-Oriented Text Detection in Natural Images.
- Discriminatory. The selected features bear enough information to distinguish well between patterns when used to train the classifier.

The Bag of Words Model

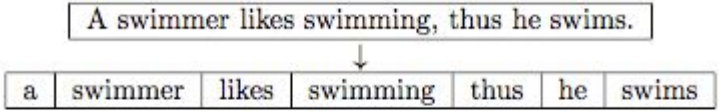
It creates a list of all unique words present across all the documents and then counts the frequency of each of these words appearing in the documents.

- vocabulary — the collection of all different words that occur in the training set and each word is associated with a count of how it occurs.

The vocabulary can then be used to construct the d-dimensional feature vectors for the individual documents where the dimensionality is equal to the number of different words in the vocabulary (
d=|V|). This process is called vectorization.

Tokenization

Tokenization describes the general process of breaking down a text corpus into individual elements that serve as input for various natural language processing algorithms. Usually, tokenization is accompanied by other optional processing steps, such as the removal of stop words and punctuation characters, stemming or lemmatizing, and the construction of n-grams.



A swimmer likes swimming, thus he swims.

↓

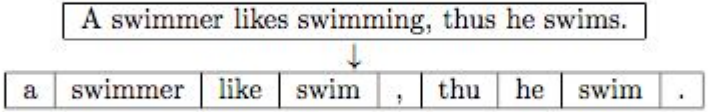
a	swimmer	likes	swimming	thus	he	swims

Stop Words

Stop words are words that are particularly common in a text corpus and thus considered as rather un-informative (e.g., words such as so, and, or, the, ...”). One approach to stop word removal is to search against a language-specific stop word dictionary. An alternative approach is to create a stop list by sorting all words in the entire text corpus by frequency. The stop list — after conversion into a set of non-redundant words — is then used to remove all those words from the input documents that are ranked among the top n words in this stop list.

Stemming and Lemmatization

Stemming describes the process of transforming a word into its root form.

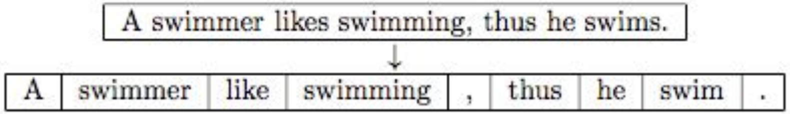


A swimmer likes swimming, thus he swims.

↓

a	swimmer	like	swim	,	thu	he	swim	.

Stemming can create non-real words, such as “thu” in the example above. In contrast to stemming, lemmatization aims to obtain the canonical (grammatically correct) forms of the words, the so-called lemmas. Lemmatization is computationally more difficult and expensive than stemming, and in practice, both stemming and lemmatization have little impact on the performance of text classification



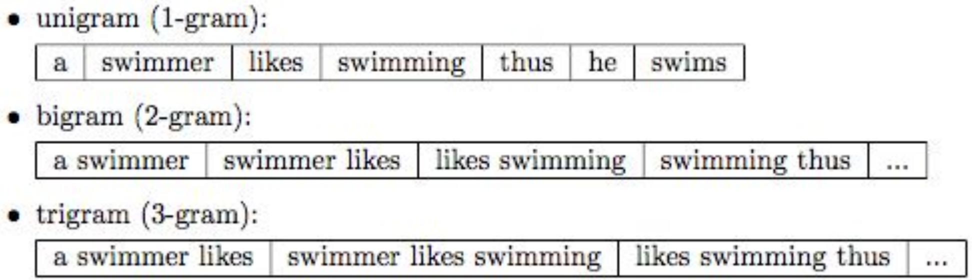
A swimmer likes swimming, thus he swims.

↓

A	swimmer	like	swimming	,	thus	he	swim	.

N-Grams

In the n-gram model, a token can be defined as a sequence of n items. The simplest case is the so-called unigram (1-gram) where each word consists of exactly one word, letter, or symbol. All previous examples were unigrams so far. Choosing the optimal number n depends on the language as well as the particular application.



- unigram (1-gram):

a	swimmer	likes	swimming	thus	he	swims

- bigram (2-gram):

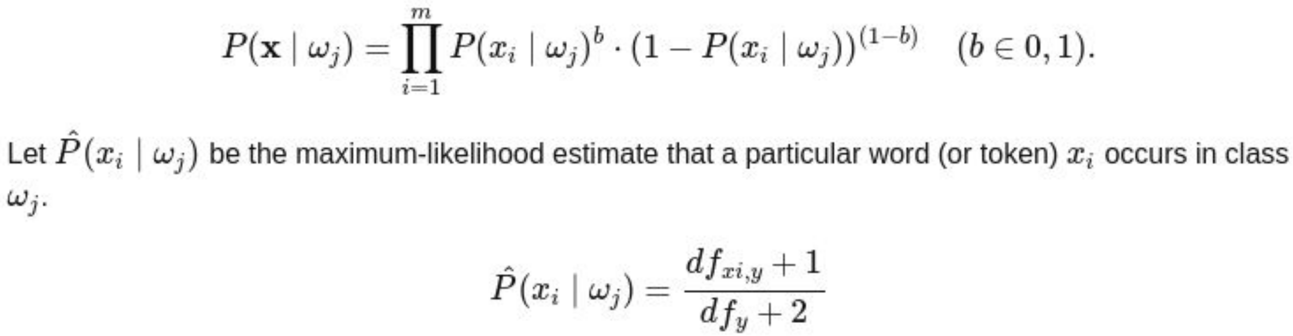
a swimmer	swimmer likes	likes swimming	swimming thus	...

- trigram (3-gram):

a swimmer likes	swimmer likes swimming	likes swimming thus	...

Multi-variate Bernoulli Naive Bayes

The Multi-variate Bernoulli model is based on binary data: Every token in the feature vector of a document is associated with the value 1 or 0. The feature vector has m dimensions where m is the number of words in the whole vocabulary (in Section The Bag of Words Model; the value 1 means that the word occurs in the particular document, and 0 means that the word does not occur in this document.



$$P(\mathbf{x} \mid \omega_j) = \prod_{i=1}^{m} P(x_i \mid \omega_j)^b \cdot (1 - P(x_i \mid \omega_j))^{(1-b)} \quad (b \in 0, 1).$$

Let $\hat{P}(x_i \mid \omega_j)$ be the maximum-likelihood estimate that a particular word (or token) x_i occurs in class ω_j.

$$\hat{P}(x_i \mid \omega_j) = \frac{df_{xi,y} + 1}{df_y + 2}$$

Multinomial Naive Bayes

A alternative approach to characterize text documents — rather than binary values — is the term frequency (tf(t, d)). The term frequency is typically defined as the number of times a given term t (i.e., word or token) appears in a document d (this approach is sometimes also called raw frequency). In practice, the term frequency is often normalized by dividing the raw term frequency by the document length.

$$\hat{P}(x_i \mid \omega_j) = \frac{\sum tf(x_i, d \in \omega_j) + \alpha}{\sum N_{d \in \omega j} + \alpha \cdot V}$$

Thank You