INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY

HYDERABAD

Introduction to NLP

Prof Manish Shrivastava

Academic Year 2020 - 2021

Unsupervised Chunking for Indian Languages

Author
Chayan Kochar
Tanishq Goel

Abstract

Chunking, or shallow-parsing, is a task that requires the identification of syntactic units which belong together, for example verbs and verbal auxiliaries are one chunk. However, it does not specify their internal structure, nor their role in the main sentence. The aim of this project is to create a phrase based chunking algorithm for Indian Languages.

We have implemented chunking of Indian Languages using unsupervised approach. Although supervised learning algorithms have resulted in the state of the art and high accuracy systems on varieties of tasks in the NLP domain, the performance in source-poor language is still unreasonable. A fundamental obstacle of statistical shallow parsing for the quantities of world's language is the shortage of annotated training data. Furthermore, the work of well-understand hand annotation has proved to be expansive and time consuming. Hindi Language still has relatively decent amount of training data but that can't be said about languages like *pali* or *prakrit*

We have implemented chunking through K-Means Clustering which is a very popular unsupervised ML Algorithm which make inferences from data set using only input vectors without referring to known or labelled outcomes. Main objective of K-means is simple: group similar data points together and discover underlying patterns. Algorithm is discussed in detail further.

Chapter 1

Introduction

1.1 Shallow Parsing

Shallow parsing, also known as light parsing or chunking, is a technique for analyzing the structure of a sentence in-order to identify these phrases or chunks. We start by first breaking the sentence down into its smallest constituents (which are tokens such as words) and then grouping them together into higher-level phrases.

1.2 Unsupervised Approach

Unsupervised Learning is a machine learning technique in which the users do not need to supervise the model. Instead, it allows the model to work on its own to discover patterns and information that was previously undetected. It mainly deals with the unlabelled data. It allow users to perform more complex processing tasks compared to supervised learning. Although, unsupervised learning can be more unpredictable compared with other natural learning methods. Unsupervised learning algorithms include clustering, anomaly detection, neural networks, etc.

1.3 K-Means Clustering

K-means clustering is one of the simplest and popular unsupervised machine learning algorithms. Typically, unsupervised algorithms make inferences from data sets using only input vectors without referring to known, or labelled, outcomes.

The main objective of K-means is simple: group similar data points together and discover underlying patterns. To achieve this objective, K-means looks for a fixed number (k) of clusters in a data set. A cluster refers to a collection of data points aggregated together because of certain similarities. We defined a target number k, which refers to the number of centroids we needed in the data set. A centroid is the imaginary or real location representing the center of the cluster. Every data point is allocated to each of the clusters through reducing the in-cluster sum of squares. In other words, the K-means algorithm identifies k number of centroids, and then allocates every data point to the nearest cluster, while keeping the centroids as small as possible. The 'means' in the K-means refers to averaging of the data; that is,

finding the centroid.

How the K-means algorithm works

To process the learning data, the K-means algorithm in data mining starts with a first group of randomly selected centroids, which are used as the beginning points for every cluster, and then performs iterative (repetitive) calculations to optimize the positions of the centroids. We chose clusters on the basis of verbs or nouns present in the sentence. It halts creating and optimizing clusters when either:

1. The centroids have stabilized — there is no change in their values because the clustering has been successful.
2. The defined number of iterations has been achieved.

1.4 Hierarchical Clustering

Hierarchical clustering involves creating clusters that have a predetermined ordering from top to bottom. For example, all files and folders on the hard disk are organized in a hierarchy. There are two types of hierarchical clustering, Divisive and Agglomerative.

We implemented it using Agglomerative Algorithm. In agglomerative or bottom-up clustering method we assign each observation to its own cluster. Then, compute the similarity (e.g., distance) between each of the clusters and join the two most similar clusters. Finally, repeat steps 2 and 3 until there is only a single cluster left. Before any clustering is performed, it is required to determine the proximity matrix containing the distance between each point using a distance function. Then, the matrix is updated to display the distance between each cluster. The following three methods differ in how the distance between each cluster is measured.