- Information Extraction has many applications, including business intelligence, resume harvesting, media analysis, sentiment detection, patent search, and email scanning.
- It begins by processing a document using several of the procedures first, the raw text of the document is split into sentences using a sentence segmenter, and each sentence is further subdivided into words using a tokenizer. Next, each sentence is tagged with part-of-speech tags, which will prove very helpful in the next step, named entity recognition. In this step, we search for mentions of potentially interesting entities in each sentence. Finally, we use relation recognition to search for likely relations between different entities in the text.
- The smaller boxes show the word-level tokenization and part-of-speech tagging, while the large boxes show higher-level chunking. Each of these larger boxes is called a chunk. Like tokenization, which omits whitespace, chunking usually selects a subset of the tokens.
- One of the most useful sources of information for NP-chunking is part-of-speech tags. This is one of the motivations for performing part-of-speech tagging in our information extraction system
- In order to create an NP-chunker, we will first define a chunk grammar, consisting of rules that indicate how sentences should be chunked. In this case
- chunking has different types and used most of time in tagging (TTS)
- POS - —- > Part of Speech
- Sometimes it is easier to define what we want to exclude from a chunk. We can define a chink to be a sequence of tokens that is not included in a chunk (reverse of chunking)
- chunk can be represented as tree or tags
- chunk structures can be represented using either tags or trees. The most widespread file representation uses IOB tags. In this scheme, each token is tagged with one of three special chunk tags, I (inside), O (outside), or B (begin). A token is tagged as B if it marks the beginning of a chunk. Subsequent tokens within the chunk are tagged I. All other tokens are tagged O. The B and I tags are suffixed with the chunk type, e.g., B-NP, IN-NP
- depends on the tagging the chunker would get better accuracy so we could use unigram parser for improving our tagging like bigram and unigram
- we need to make use of information about the content of the words, in addition to just their part-of-speech tags, if we wish to maximize chunking performance.
- The goal of a named entity recognition (NER) system is to identify all textual mentions of the named entities. This can be broken down into two subtasks: identifying the boundaries of the NE, and identifying its type
- relation of entities can be recognized by defined it with regex
- Relation extraction can be performed using either rule-based systems, which typically look for specific patterns in the text that connect entities and the intervening words; or using machine-learning systems, which typically attempt to learn such patterns automatically from a training corpus.