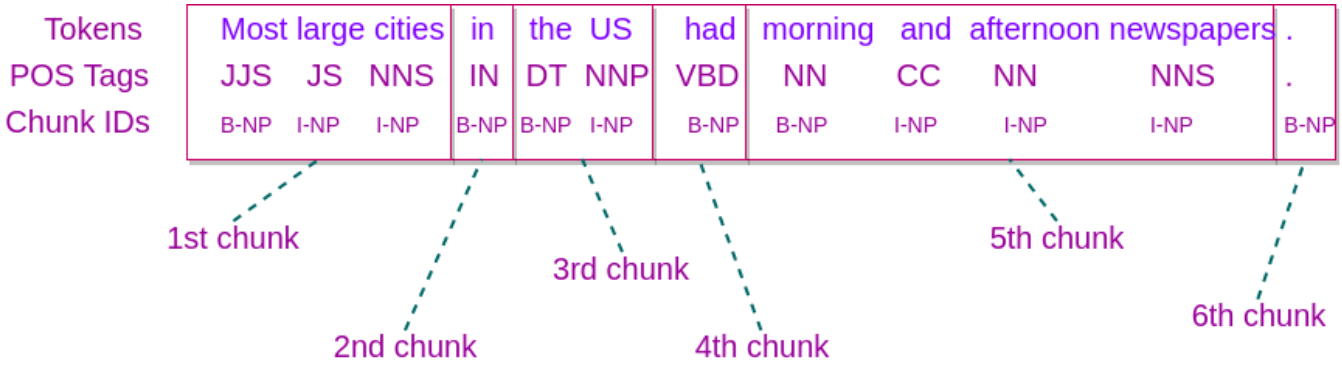
Chunker Example in Apache OpenNLP

What does a Chunker do ?

A chunker breaks the sentence into groups(of words) containing sequential words of sentence, that belong to a noun group, verb group, etc.

In this section Apache OpenNLP Tutorial we shall write a java program to demonstrate the usage of Chunker API with the help of ChunkerME class for chunking (NLP task). Also we shall analyze the output (chunks) and what the chunks represent.

Pictorial representation of the test sentence that we are going to divide into chunks is given below :



Chunker Example in Apache OpenNLP

Java Program : Chunker Example in Apache OpenNLP

Chunker API needs tokens and corresponding pos tags of a sentence. In this example program, we shall use provide the takens as an array (you may use Tokenizer for this job), and a POS Tagger to postag the tokens. And then both the tokens and postags go as input to chunker. Please follow the below program with well written comments for better understanding.

ChunkerExample.java

```
import
```

```
import opennlp.tools.chunker.ChunkerME;
import opennlp.tools.chunker.ChunkerModel;
import opennlp.tools.lemmatizer.DictionaryLemmatizer;
import opennlp.tools.postag.POSModel;
import opennlp.tools.postag.POSTaggerME;

import java.io.*;

/**
* Chunker Example in Apache OpenNLP
```

```
*/
public class ChunkerExample {

   public static void main(String[] args){
      try{
         // test sentence
         String[] tokens = new String[]{"Most", "large", "cities", "in", "the", "US", "had",
               "morning", "and", "afternoon", "newspapers", "."};

         // Parts-Of-Speech Tagging
         // reading parts-of-speech model to a stream
         InputStream posModelIn = new FileInputStream("models"+File.separator+"en-pos-maxent.bin");
         // loading the parts-of-speech model from stream
         POSModel posModel = new POSModel(posModelIn);
         // initializing the parts-of-speech tagger with model
         POSTaggerME posTagger = new POSTaggerME(posModel);
         // Tagger tagging the tokens
         String tags[] = posTagger.tag(tokens);

         // reading the chunker model
         InputStream ins = new FileInputStream("models"+File.separator+"en-chunker.bin");
         // loading the chunker model
         ChunkerModel chunkerModel = new ChunkerModel(ins);
         // initializing chunker(maximum entropy) with chunker model
         ChunkerME chunker = new ChunkerME(chunkerModel);
         // chunking the given sentence : chunking requires sentence to be tokenized and pos tagged
         String[] chunks = chunker.chunk(tokens,tags);

         // printing the results
         System.out.println("\nChunker Example in Apache OpenNLP\nPrinting chunks for the given sentence...");
         System.out.println("\nTOKEN - POS_TAG - CHUNK_ID\n-------------------------");
         for(int i=0;i< chunks.length;i++){
            System.out.println(tokens[i]+" - "+tags[i]+" - "+chunks[i]);
         }
      } catch (FileNotFoundException e){
         e.printStackTrace();
      } catch (IOException e) {
         e.printStackTrace();
      }
   }
}
```

Output :

Program Output

```
Printing chunks for the
given sentence
```

```
Printing chunks for the given sentence...

TOKEN - POS_TAG - CHUNK_ID
-------------------------
Most - JJS - B-NP
large - JJ - I-NP
cities - NNS - I-NP
in - IN - B-PP
the - DT - B-NP
US - NNP - I-NP
had - VBD - B-VP
morning - NN - B-NP
and - CC - I-NP
afternoon - NN - I-NP
newspapers - NNS - I-NP
. - . - O
```

Let us see what these chunks (displayed in the output) represent.

If you observe, there are two notations for the chunk_id s in the output.

- B- : Represents the **start** of a chunk
- I- : Represents the **continuation** of a chunk

We shall represent the output in a table, and mention the chunks in the last column.

Token	POS Tag	Chunk ID	Chunk
Most	JJS	B-NP	**1st chunk** in the sentence (Noun Phrase)
large	JJ	I-NP	
cities	NNS	I-NP	
in	IN	B-NP	**2nd chunk** in the sentence (Noun Phrase)
the	DT	B-NP	**3rd chunk** in the sentence (Noun Phrase)
US	NNP	I-NP	
had	VBD	B-NP	**4th chunk** in the sentence (Noun Phrase)
morning	NN	B-NP	**5th chunk** in the sentence (Noun Phrase)
and	CC	I-NP	
afternoon	NN	I-NP	
newspapers	NNS	I-NP	
.	.	0	no chunk

Hence, the sentence has been divided into five chunks. In this example we have only -NP (Noun Phrase). There are other phrases like -PP(Preposition Phrase), -VP(Verb Phrase), etc. Try out with different sentences and observe the chunks.

Official Manual for chunker is present at [https://opennlp.apache.org/docs/1.8.0/manual/opennlp.html#tools.parser.chunking.api]

Conclusion :

We have learnt what a chunker does, and how to use the Java Chunker API in Apache OpenNLP, and how to identify the start and continuation of a chunk, different types of chunks (-NP, -VP, -PP,..)