ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 7, July 2015

A Review on Text Chunker for Punjabi Language

Ubeeka Jain[1], Jasbir Kaur[2]

Assistant Professor, R.I.E.I.T , Railmajra, Punjab[1]

M.Tech Research Scholar, R.I.E.I.T , Railmajra, Punjab[2]

Abstract: Parsing is the process of assigning a parse tree to the sentence. There are many problems related to the process of full parsing. Shallow parsing or chunking is the alternative for full parsing. In chunking the phrases of the sentences are chunked together. Chunking is more efficient and robust as it takes less time and always gives a solution. It is often deterministic as it gives only one solution to a problem. Chunkers are used in a large no. of NLP applications. Such as information extraction, named entity recognition, spell checkers, search etc . Chunkers are relatively difficult to build for Indian languages as there arise many problems during the system development. Chunkers identify the noun or verb chunks. Chunks are the non overlapping regions. In this work text chunker of Punjabi language is built and the greedy based algorithm is used for the machine learning and training of data set.

Keywords: Natural language Processing (NLP), Part of Speech Tagger(POS), Punjabi chunker.

I. INTRODUCTION

In NLP Computers are used to understand and manipulate text and speech to do some useful work NLP is the branch of Computer science mainly dealing with developing of systems by which computers can interact with human using natural language . NLP includes various computational and analyzing processes which enable machine to understand the language. Punjabi is an **Indo-Aryan** language. It is the 10TH most spoken language in the world and native language of about 131 million people. Most of the Punjabi speaking people live in Punjab region of Pakistan and India. It is also spoken in Himachal Pradesh, Haryana and Delhi and many countries in abroad. Punjabi is written in two different scripts called **Gurmukhi** and **Shahmukhi.**

Some of the applications for NLP are Part of Speech tagging (POS), Question Answering system, Name Entity Recognition (NER), and Multiple Word Expression (MWE) etc. which are used in machine translation.

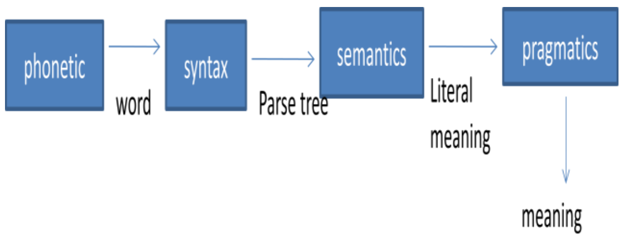
Chunking: hunking is the process of dividing the sentence into chunks. Chunks are the non-overlapping regions in a sentence. Chunks are correlated group of words[1].

The phrase chunker divides the sentence into noun phrases or verb phrases. These phrases are grouped together i.e. all the verbs occurring in a sentence are chunked in a single chunk and all the noun phrases are grouped in another single chunk. There also exist adjective phrases and noun adverb phrases.[2]

There are many levels of language analysis. These are shown in the following figure. The parsing phase lies in the syntax level of language analysis. Parsing is the process of generation of parse tree for a sentence.

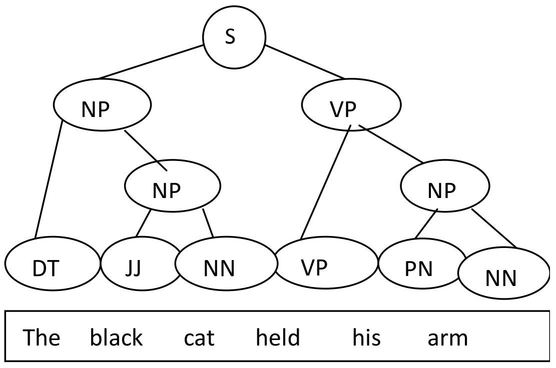
Chunking is the alternative to parsing. There exists no complete grammar for any language. Ambiguity exists for many sentences. Ambiguity is the generation of more than one parse tree for one sentence. Full parsing takes a reasonable time for large amount of data. Chunking is more efficient and robust as it takes less time and always gives a solution. It is often deterministic as it gives only one solution to a problem. Context is Small and local.

It can be applied to very large text resources i.e. web.[3]



The output of the chunker consists of series of non – overlapping regions that are also non recursive and do not contain each other. Thus the output of chunker is different from the parsing and it is easier as compared to parsing.

Lets take an example :
The black cat held his arm.
The chunker will give the output:
[NP the black cat][VP held his arm]
But The parser will generate the parse tree as follows:



Thus, In identifying the part of speech tags and generating a full parsed tree of it ,chunker work as a middle step.
Chunking can be useful for many applications of NLP including information extraction, spell checkers, named entity recognition. As the whole chunk is used. As in question answering system any of the chunk can be used as the part of the answer.

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 7, July 2015

the sentence "*He reckons the account deficit will narrow to only # 1.8 billion in September.*" Can be chunked as by enclosing the chunks in square brackets.
[NP He] [VP reckons] [NP the account deficit] [VP will narrow] [PP to] [NP only # 1.8 billion] [PP in] [NP September].
The tag next to the open bracket denotes the type of the chunk.

The chunks can be represented using two notations:
A. *Tag representation:*
IOB tags: each token is tagged with one of three special chunk tags, "INSIDE", "OUTSIDE", or "BEGIN"
1) A token is tagged as "BEGIN" if it is at the beginning of a chunk, and contained within that chunk
2) Subsequent tokens within the chunk are tagged "INSIDE"
3) All other tokens are tagged "OUTSIDE".

B. *Tree representation:* trees spanning the entire text[3]

For Indian Languages the chunk tagset is:

Sl. No	*Chunk Type*	*Tag Name*
1	Noun Chunk	NP
2.1	Finite Verb Chunk	VGF
2.2	Non-finite Verb Chunk	VGNF
2.3	Infinitival Verb Chunk	VGINF
2.4	Verb Chunk (Gerund)	VGNN
3	Adjectival Chunk	JJP
4	Adverb Chunk	RBP
5	Chunk for Negatives	NEGP
6	Conjuncts	CCP
7	Chunk Fragments	FRAGP
8	Miscellaneous	BLK

[4]
chunking phase comes after the part –of-speech tagging phase. In POS phase each word of the sentence is given a tag.
open classes -- noun, verb, adjective, adverb
closed classes -- prepositions, determiners, conjuctions, pronouns, particples
Tagging is done manually or by using some tool. Then this tagged data is analysed and chunking is done.

TECHNIQUES :
There are various techniques for chunk parsing implementation:
1) *Regular expression matching*: it defines a regular expression that matches a sequence of tags
2) *Chinking:* the text in the sentence that is not a chunk is taken as chink
3) *Finite state transducer:* it is an efficient machine that adds brackets to the text
4) *Transformational regular expression:* to add brackets to a string of tags it defines the regular expression for transformation

Rest of the paper is organized as follows the section 2 describes the applications using chunker. Section 3 briefs the literature survey about the related work done on chunking in various languages. Section 4 describes the problems related to previous systems and the objectives and methodology of the proposed system. Section 5 consists of conclusion and future scope of the proposed system

II. CHUNKER APPLICATIONS

Chunkers are used as a resource component for many NLP applications.
A. *Information extraction*: the chunker divides the sentence into chunks of interrelated data. Noun phrase and verb phrase are chunked and can be used in information extraction systems. IE focuses on discovering names of people and events they participate in, from a document.
B. *Question Answering system*: the complete chunk can be used as the answer of the question asked. question-answering provides the user with either just the text of the answer itself or answer-providing passages.
C. *Spell Checkers*: checks the wrongly typed words within the sentence.
D. *Named entity identification*: in this system the main aim is to identify the particular words in the document. Such as people , places and other nouns in the sentence.
E. *Search*: searching of a particular noun or verb can be done. As the sentence is chunked in pieces, search becomes an easy task and the whole chunk can be represented as the search result
F. *Machine translation*: machine translation is the process of translating one language into another language. Chunking is useful in this task as the chunks are converted into another language.

III. LITERATURE SURVEY

The first work in chunking is done by Church on English language. He performed machine based learning(Church K, 1988) . Transformation based learning approach was used for English language by Ramshaw and Marcus in 1995[5]. HMM based tagging method was used for chunking by Skut and Brants in1998[6]. Zhou in 2000[7] used the HMM method and achieved the recall and precision of 92.25 and 91.99 respt. Support vector machines SVM for chunking is used by Kudo and Matsumoto in 2001[8] for English. Memory based phrase chunking is used by Veenstra and Bosch in 2000. Osborne [9] in 2000 experimented with various techniques for the process of chunking

[5] in 1995, Ramshaw and Marcus used the transformation based approach for English language. He

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 7, July 2015

took two tagsets for his experiment. The chunking is done as the tagging is done in transformation based learning approach. As the training data wall street journal articles from Treebank are used. The size of training data has significant impact on the results. a precision of 91.8% is obtained and a recall of 92.3% for base np chunks when trained on 200000 words using lexical and POS information is obtained.

[6] **Brants** in 1999 presented a method based on Marcov models for partial parsing. Testing was performed on 300000 words taken from the NEGRA corpus consisting of German newspaper texts. Recall of 54% for 1 layer and 84.8% for 9 layers; precision of 91.4% for 1 layer and 88.3% for 9 layers is obtained.

In **CoNLL 2000** task 11 types of phrases are taken into account for performance evaluation:
1) Noun Phrase (NP)
2) Verb Phrase (VP)
3) Prepositional Phrase (PP)
4) Adverb Phrase (ADVP)
5) Adjective Phrase (ADJP)
6) Subordinated Clause (SBAR)
7) Conjunction Phrase (CONJP)
8) List Markers (LST)
9) Interjections (INTJP)
10) Particles (PART)
11) Unlike Coordinated Phrases (UCP)

[9]**Osborne in 2000** experimented with various techniques for chunking. He experimented on a no. of shallow parsers and trained them by using artificial training data containing noise. He wanted to check that what happens if the training data is noisy.[12] He found that the shallow parsers are robust and only a large amount of noisy data can impact on the performance. No one particular technique is efficient to cope up with the degradation performed by noise.

[10]Tjong Kim Sang in 2002 apply memory based learning to shallow parsers. Weakness of MBL is identified that it can have difficulty in identifying a large no of features. bidirectional hill climbing method for the feature selection was proposed. [11] he also found that the majority voting and stacking can increase the performance.

[13] **Jisha P Jayan and Rajeev R R** used statistical approach for chunker for Malayalam. This approach uses bi-gram, tri-gram and n- gram. HMM based approach is used in the experiment. In the training of system 15245 tokens are used. Result on comparing 200 tokens is:
Equal : 184/200 (92.00%) Different : 16/200 (8.00%) the system gives about 92% of accuracy

[14] Dhanalakshmi V made the chunker on tamil language. He has used SVM based machine learning for tamil language chunker. He has used 9 tags for chunking and generate his own tagset for training and testing. 95.82% of the accuracy is obtained. yamcha is an open source text chunker used for chunking. 225000 words for training and testing are used.

[15]Akshey singh, sushma made the HMM based text chunker for hindi language. They tried several methods to classify the chunks and found that a rule based approach gave the best results.
92.63% for chunk boundary identification task and 91.70% for the composite task of chunk labeling with a recall of 100% is obtained.

[16] Dipanjan Das, Monojit Choudhury made the chunker for Bengali language based on on the aforementioned framework. They used 39 tags. 30000 words for testing are used. The precision and recall rates are 96.12% and 98.03% are obtained. An Affinity Based Greedy Approach towards Chunking is used.

IV. PROBLEM STATEMENT AND PROPOSED SOLUTION

PROBLEM STATEMENT:

Chunkers are relatively difficult to build for Indian languages as there arise many problems during the system development. Chunkers identify the noun or verb chunks. Chunks are the non overlapping regions. There are chunkers available for various Indian languages like hindi, Bengali etc. but till now there is no chunker available for Punjabi language. There are many reasons behind it. First of all, punjabi is a resource poor language. All the work done in this language is merely at the first level of the development. POS taggers are developed for Punjabi. But all the further developments are based on chunker which is not available yet. Machine learning requires a large amount of data. Training of the data set is difficult. Lexical ambiguity is present, a word can belong to different category.

PROPOSED SOLUTION:

In objective of our work is first of all identify the part-of-speech tagged data, which is done either manually or by using the already developed POS tagger. Then we will find the chunks in the tagged data by taking the help of any linguistic. Next task is machine learning by the training data. Then the development of the algorithm takes place. We will use the greedy based algorithm. In it largest chunk is identified and then there is conflict present in the type of the chunk. E.g. in some cases the chunk is NP and in other case chunk is VP. Then the frequency of the occurrence of the word is analyzed. And the chunk is tagged accordingly. At the end, the results obtained are compared with the other chunkers if available.

V.CONCLUSION

Chunkers are developed after part-of-speech tagging phase and are the foundation of many natural language processing applications like spell checkers, machine translation, information extraction, question answering system etc. we shall develop a resource for the further development in the Punjabi language.

REFERENCES

1) Steven P. Abney. Parsing by Chunks. Kluwer, principle-based parsing: computation and psycholinguistics edition, 1991.
2) Anil Kumar Singh, (2008) Language Technologies Research Centre, IIIT, Hydrabad India, NLP for Less Privileged Languages:

Where do we come from? Where are we going? In IJCNLP Workshop on NLP

3) Claire Gardent,CNRS/LORIA.Campus Scientifique,BP 239,F-54 506 Vandoeuvre-l`es-Nancy, France
4) Guidelines For POS And Chunk Annotation For Indian Languages by A. Bharati, D. M. Sharma, Lakshmi Bai, Rajeev Sangal Language Technologies, IIT, Hyderabad
5) L. A. Ramshaw and M. P. Marcus. Text chunking using learning based on transformation, 1995.
6) Thorsten Brants. Cascaded markov models. 9th Conference of the European Chapter of the Association for Computational Linguistics (EACL-99), Bergen, Norway, 1999.
7) GuoDong Zhou, Jian and TongG. Tey. Hybrid text chunking. CoNLL-2000 and LLL-2000
8) Kudo and Matsumoto, use of support vector learning for chunk identification, 2000
9) Miles Osborne. Shallow parsing using noisy and non-stationary training material 2002.
10) E. Tjong Kim Sang and S. Buchholz. the CoNLL-2000 shared task
11) Erik F. Tjong Kim Sang, sabine Text chunking by system combination. CoNLL-2000 and LLL-2000,
12) Miles Osborne. Shallow parsing as part-of-speech tagging. CoNLL-2000 and LLL-2000,
13) Parts Of Speech Tagger and Chunker for Malayalam – Statistical Approach by Jisha P Jayan, Rajeev R R Department of Tamil University Thanjavur
14) POS Tagger and Chunker for Tamil Language Dhanalakshmi V1, Anand kumar M1, Rajendran S2, Soman K P1
15) HMM Based Chunker for Hindi Akshay Singh, Sushma Bendre IIIT Hyderabad, India.
16) An Affinity Based Greedy Approach towards Chunking for Indian Languages Dipanjan Das, Monojit Choudhury, Department of Computer Science and Engineering Indian Institute of Technology, Kharagpur.