www.ijecs.in
International Journal Of Engineering And Computer Science ISSN:2319-7242
Volume 6 Issue 7 July 2017, Page No. 22121-22127
Index Copernicus value (2015): 58.10 DOI: 10.18535/ijecs/v6i7.36

Punjabi Chunker using Bootstrapping Approach

Sunanda[1], Ubeeka Jain[2]

[1]M.tech research scholar, R.I.E.I.T , Railmajra ,Punjab
[2]Associate Professor, I.K. Gujral Punjab Technical University, Rayat Institute of Engineering and Information Technology, India
[1]chandelsunanda@gmail.com, [2]ubeekajain@gmail.com

ABSTRACT: This paper represents Punjabi Chunker using bootstrapping approach. Bootstrapping is an approach which does not need any external input that's why it is also known as self-starting process. It is semi-supervised technique in which collection of both labeled and unlabeled data is taken. It helps to make use of unlabeled data by training a small amount of labeled data. Semi supervised learning is fall in the middle of supervised learning and unsupervised learning Chunking is the process of breaking long strings of information into units or chunks. Chunking is different from parsing. POS, Named entity Recognition and sentence breaking are the main applications of NLP in which chunking are used. This research work is different from greedy algorithm because in this approach both labeled (trained) and unlabeled data set is used to built text Chunker for Punjabi language.

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

1. Introduction

NLP is an area of research and application that explore how computers can be used to understand and manipulate the human language, text or speech to perform desired task. It is a branch of computer science that deals with the human computer interaction. NLP automates the translation process between computers and humans.

Punjabi is written in two different scripts called **Gurumukhi** 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), Sentence breaking etc. which are used in machine translation.

Chunking: Chunking is the process of breaking long strings of information into units or chunks. Chunking is also known as shallow parsing. Chunking is an analysis of sentence which identifies the constitute parts that is Noun group (NG) and verb group (VG) and then links that chunks into the grammatical meaning. Chunks are the non-overlapping regions in a sentence. chunks are non-exhaustive as some words of a sentence is not grouped into a chunk. Chunks are correlated group of words[1].

The phrase Chunker divides the chunks 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]

Chunking is popular alternative to parsing. It is the context of tagging and also known as shallow parsing or robust 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 than full parsing as it takes less time and always gives a deterministic solution. Context is Small and local. it can be applied to very large text resources i.e. web.[3]

The output of Chunker is different from the full parsing.

Example: I saw the big dog on hill.

Chucker's Output: [NP I][VP saw] [NP the big dog].

The tag next to the open bracket denotes the type of the chunk.

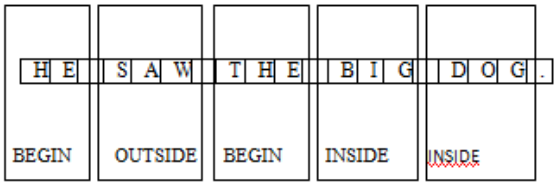
The Chunks can be represented using two notations:

DOI: 10.18535/ijecs/v6i7.36

A. Tag representation:

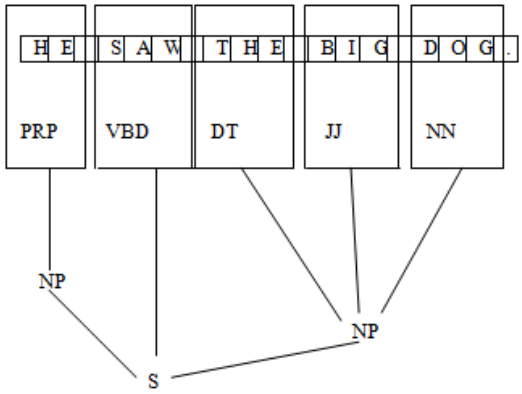
There are three types of IOB tags:

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]

Chunking phase comes after the part –of-speech tagging phase. In POS phase each word of the sentence is given a tag.



Tagging is done manually or by using some tool. Then this tagged data is analyzed and chunking is done.

2. Literature Review:

[7] **Shlomo Argamon-englosen, et al. (1999)** presents a Novel memory based learning method that identifies shallow patterns in new text based on a bracketed training body of text. Generalization is performed On-line at recognition time. This paper presents investigate results for recognizing noun phrase, subject-verb phrase and verb object in English language.

[13] Roald Eiselen, developed the chunker for ten south Indian languages. He defines the development of protocols, annotated phrase chunking data sets and automatic phrase chunkers for ten South African languages. In this CRF based chunker were created and tested. 15000 of phrase chunk annotated tokens were developed in this system. He attained f-scores 93%.

[11] Dipanjan Das et al. they developed chunker for Bengali language. In this paper a computational framework for chunking based on valency theory and feature structures has been described. Valency theory focuses on the anlayazation of sentences and taget the certain type of that word in the sentence. For example many of words are verb and others are optional that means they can be optional. This chunker for Bengali depends upon the morphological analyzer and POS tagger and they both attained accuracy of 95% and 91% respectively.

[15] Curtis M. Kularski (2010) this paper is an exploration of the Experimental process of chunking. Three research studies on the topic of chunking will be analyzed to Present on the topic and provide perspective on the indications of chunking on the overall storage and retrieval system of the brain.

[10] Dinesh Kumar et al. (2010) this paper describes about the Part of Speech (POS) taggers justified for various Indian Languages like Hindi, Punjabi, Malayalam, Bengali and Telugu. Various part of speech tagging approaches like Hidden Markov Model (HMM), Support Vector Model (SVM), Rule based approaches, Maximum Entropy (ME) and Conditional Random Field (CRF) have been used for POS tagging.

[12] Rijuka Pathak et al. (2014) they worked on Chhattisgarhi and other Indian state languages. This is the main language spoken by 1.5 million people in the world. The main objective of this paper is to develop machine learning, to develop translator, to develop dictionary and to develop Pos tagger. There are many type of POS developer in this paper we will see different kind of POS tagger.

[8] Chandan Mittal et al 2015: They had followed Hidden Markov Model in achieving their goal. They used Viterbi Algorithm for calculating the highest probability of chunks and to train the system, Baum-Welch algorithm was followed

and 25,000 lines of chunked Punjabi text were used. An annotated text file having 1,000 lines was used for testing the system. The accuracy of the system to find the chunk boundaries of the system is about 80% approx and the labeling is applied with an accuracy of about 98% and the labeling is applied with an accuracy of about 82%.

[1] Ubeeka jain and jasbir kaur in 2015: In this research, first standardized text chunker for Punjabi language is created and the greedy based algorithm is used for the machine learning and training of data set. This is the first chunker for Punjabi language in natural language processing. They used greedy algorithm for chunking. In this technique only supervised learning approach is used. In which all the data is labeled. Results of this research are very efficient as precision 93%, Recall 75% and F-measure is 83%.

S.no.	Chunk	Chunk Description
1	_NP	Noun chunk
2	_CCP	Conjunction chunk
3	_VGF	Verb chunk
4	_RBP	Adverb chunk
5	_JJP	Adjective chunk
6	_VGNF	Verb Infinite
7	_BLK	Bulk Phrase

[14] Biplav Sarma and Anup Kumar Barman in 2015: they did comprehensive survey of noun phrase chunking in natural languages. They worked for assamese language which is the native language of assam. The goal of this NP chunker is to find out the noun phrase from the text. Chunking is the process of tagging the word. This process can be used as a rapid and reliable processing phase for parsing either partial or full..

[9] Manish Shrivastava and Pushpak Bhattacharyya in 2008: In this paper, they describe a simple HMM based POS tagger, which exert a naïve (longest suffix matching) stemmer as a pre-processor to accomplish reasonably good efficiency of 93.12%. This process does not need any linguistic resource aside from a list of possible suffixes for the language. This list can be easily designed using existing machine learning techniques. The goal of this approach is to exhibit that even without employing tools like morphological analyzer or ability like a pre-compiled structured lexicon; it is achievable to tackle the morphological richness of Indian Languages.

[5] Eric F. Tjong Kim sand and Jorn veenstra: In this research work they explained seven different data representation to minimize the problem with noun phrasing. They proved data representation had secondary impact on chunking performance. They achieved high efficiency as precision 90.7%, recall 91.1% and $F_{\beta=1}$ 90.9%.

[4] Hinrich Schfitze: He evaluated the algorithm on Brown corpus. This algorithm classifies the word tokens in context rather than word type. This research work describes an algorithm for tagging text (word) whose part-of-speech properties are unknown. The accuracy of this algorithm is Precision 83%, recall 78% and f-measure 79%.

3. Tag set for chunking:

There are mainly seven tags which are used in chunking. These are based on grammatical and syntactical category. Chunks are represented in square bracket and by the right hand side tag set are mentioned.

Examples of chunk descriptions are:

1. **Noun chunk:**

- [[ਗੋਰਨਗੋਰੋ\N_NNP ਨੈਸ਼ਨਲ\N_NNP ਪਾਰਕ\N_NNP ਤੋਂ\PSP]]_NP
- [[ਬਾਰ\N_NN]]_NP

2. **Conjunction chunk:**

- [[ਅਤੇ\CC_CCD]]_CCP
- [[ਜਦੋਂਕਿ\CC_CCS]]_CCP

3. **Verb Chunk:**

- [[ਪਰਿਵਰਤਿਤ\V_VM ਹੋ\V_VM_VF ਚੁੱਕੀ\V_VM_VF ਹੈ\V_VAUX]]_VGF
- [[ਸੀ\V_VAUX]]_VGF

4. **Adverb Chunk**:

- [[ਸਿਰਫ\QT_QTF]]_RBP

- [[ਦੂਰ\RB]]_RBP

5. Adjective chunk:

- [[ਲੰਬੇ\JJ]]_JJP
- [[ਉਤਪੱਤੀ\JJ]]_JJP

6. Verb Infinite:

- [[ਪਹੁੰਚਾਉਣ\V_VM_VNF]]_VGNF
- [[ਕੇ\V_VM_VNF]]_VGNF

7. Bulk Phrase:

- [[ਦੀ\PSP]]_BLK

4. Corpus Development:

Corpus is developed for the training and testing of data. Test data is trained by the corpus and then this data is updated in the system. This trained data is then ready for chunking. The sample of training data is as follows:

[[ਮਨਆਰਾ\N_NNP ਪਾਰਕ\N_NNP ਵਿਚ\PSP]]_NP

[[ਕੁਝ\QT_QTF ਹੀ\RP_RPD]]_BLK

[[ਹੀ\RP_RPD]]_BLK

[[ਇਹਨਾਂ\PR_PRP ਦੀ\PSP ਸ਼ਕਤੀ\N_NN]]_NP

[[ਜਦੋਂਕਿ\CC_CCS]]_CCP

[[ਕਿ\CC_CCS]]_CCP

[[ਛੁਪੀ\V_VM_VF ਹੁੰਦੀ\V_VM_VF ਹੈ\V_VAUX]]_VGF

[[|\RD_PUNC]]_BLK

[[ਸੂਰ\N_NN ਇਹਨਾਂ\PR_PRP ਦੰਦਾਂ\N_NN ਦੇ\PSP ਸਹਾਰੇ\N_NN]]_NP

[[ਵਾਸਤਵ\JJ]]_JJP

[[ਕਈ\QT_QTF]]_BLK

[[ਅਵਸਰਾਂ\N_NN ਤੇ\CC_CCS ਸ਼ੇਰ\N_NN]]_NP

[[ਨਾਲ\PSP ਵੀ\RP_RPD]]_BLK

[[ਟੱਕਰ\N_NN]]_NP

[[ਮਿਲੀ\V_VM_VF]]_VGF

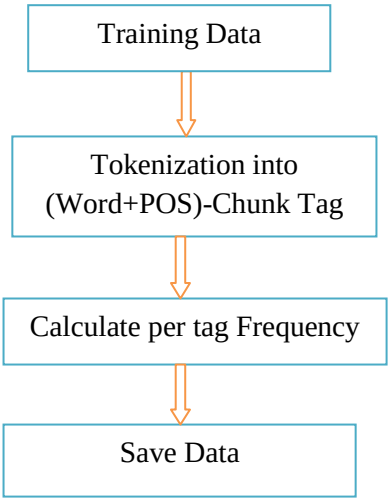
[[ਲੈਣ\V_VM_VNF]]_VGNF

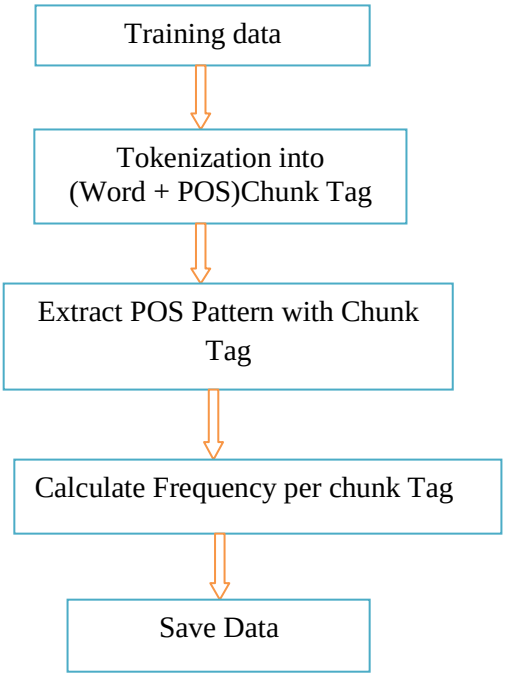
[[ਦੀ\PSP]]_BLK

5. Methodology

The overview of framework is designed below. First method is to chunk and calculate the frequency of trained data and other two methods are to chunk new data which is untrained and then find the frequency of that updated data.

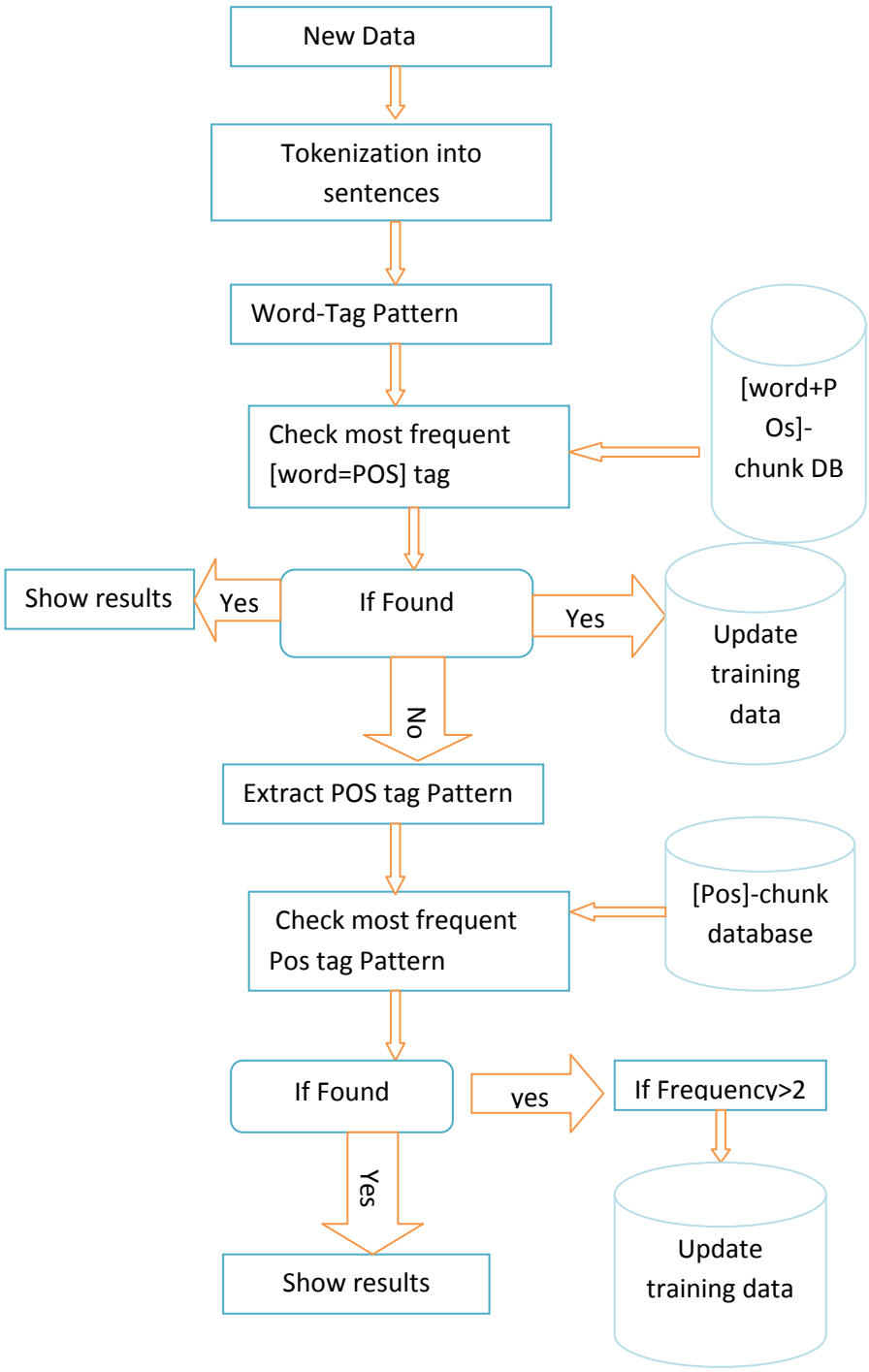
Designed flow charts are listed below:





DOI: 10.18535/ijecs/v6i7.36

3)



These flow charts describe the whole system:

In first and second flow chart system trains the data. These trained data is then ready for tokenization. After that frequency of tags are calculated.

In second flow chart first two steps are same. In this POS pattern is extracted and then frequency is calculated.

In third flow chart new data is trained. This is untrained data. This untrained data is first trained by the system, after that this updated data is saved in system. After saving the updated data POS pattern is extracted and frequency of data is calculated.

6. Result

To evaluation of the system, we have been used testing data. Testing data has been collected randomly from online news websites etc. Detail of testing data shown below:-

The accuracy of the system can be find out using these formulas:

Recall: $R = \frac{no.of\ correct\ tags\ given\ by\ system}{no.of\ possible\ correct\ answers\ in\ text}$

Precision: P = $\frac{No.of\ correct\ tags}{total\ no.of\ tags\ return\ by\ system}$

F-Measure=2* $\frac{PR}{(P+R)}$

Test Cases:

Test Set	Data Domain	Word Count
Test Set 1	Political News	5011
Test Set 2	Entertainment News	4021
Test Set 3	Sports News	3421

Evaluation of Test Cases:

Test Set	Recall	Precision	F-Measure
Test Set 1	85.33	88.45	86.85
Test Set 2	84.31	87.23	85.74
Test Set 3	76.13	78.35	77.22

Overall Accuracy of system:

S.no.	Domain	Result
1	Recall	85.33%
2	Precision	88.45%
3	f-measure	86.85%

7. Conclusion

This paper presents Punjabi chunking using bootstrapping approach. This system performs the chunking of Punjabi words into seven chunks. We are using semi supervised technique in which half of the data is labeled. This system is more efficient than the existing system. This work will motivate to future researchers for development in Punjabi language/field.

8. References

[1] Ubeeka jain, jasbir kaur, Text Chunker FOR Punjabi, International journal of Current engineering and technology(2015) 3349-3353

[2] Chandan Mittal*, Vishal Goyal and Umrinderpal Singh, HMM Chunker for Punjabi, Indian Journal of Science and Technology (2015)

[3] Manjit Kaur, Mehak Aggerwal, Sanjeev Kumar Sharma, Improving Punjabi Part of Speech Tagger by Using Reduced Tag Set, International Journal of Computer Applications & Information Technology(2015) 142-148

[4] Hinrich Schfitze, "Distributional Part-of-Speech Tagging", CSLI, Ventura Hall Stanford, CA 94305-4115, USA.

[5] Erik F.Tjong Kim Sang and Jorn Veenstra,"Representing Text Chunks",Center of Dutch Language and Speech University of Antwerp Universiteitsplein 1 B-2610 Wilrijk, Belgium and Computational Linguistics Tilburg University P.O. Box 90153 5000 LE Tilburg, The Netherlands.

[6] Lance A. Ramshaw, and Mitchell P. Marcus. (1995) Text Chunking Using Transformation-Based Learning. Proceedings of the 3rd Workshop on Very Large Corpora (1995) 88-94

[7] Shlomo Argamon-Engleson, Ido Dagan, Yuval krymolowski, "A Memory Based approach to learning shallow Natural language patterns", Department of Mathematics and Computer Science, Bar Ilan University, Israel (1999).

[8] Chandan Mittal, Vishal Goyal and Umrinderpal Singh," HMM Chunker for Punjabi", Department of computer Science, Punjab University, Patiala – 147002, India

[9] Manish Shrivastava and Pushpak Bhattacharyya," Hindi POS Tagger Using Naive Stemming : Harnessing Morphological, Information Without Extensive Linguistics Knowledge", Department of Computer Science and Engineering, Indian Institute of Technology, Bombay.

[10] Dinesh Kumar, Gurpreet Singh Josan, "Part of Speech Taggers for Morphologically Rich Indian Languages: A Survey", DAV Institute of Engineering & Technology Jalandhar, Punjab, INDIA. (2010)

[11] Dipanjan Das, Monojit Choudhury, An Affinity Based Greedy Approach towards Chunking for Indian Languages Department of Computer Science and Engineering Indian Institute of Technology, Kharagpur

[12] Rijuka Pathak, Somesh Dewangan, Natural Language Chhattisgarhi: A Literature Survey, International Journal of Engineering Trends and Technology (IJETT) (2014). 113-117

[13] Roald Eiselen," South African Language Resources: Phrase Chunking", Centre for Text Technology, North-West University, Potchefstroom Campus, South Africa.

[14] Biplav Sarma and Anup kumar Barman," A Comprehensive Survey of Noun Phrase Chunking in Natural Languages", Department of Information Technology,Gauhati University, Assam, India.

[15] In 2010, Curtis M. Kularski, "Chunking", Fayetteville State University.