Noun Phrase Chunking for Marathi using Distant Supervision

Sachin Pawar[1,2] Nitin Ramrakhiyani[1] Girish K. Palshikar[1]
Pushpak Bhattacharyya[2] Swapnil Hingmire[1,3]
{sachin7.p, nitin.ramrakhiyani, gk.palshikar}@tcs.com
pb@cse.iitb.ac.in, swapnil.hingmire@tcs.com
[1]Systems Research Lab, Tata Consultancy Services Ltd., Pune, India
[2]Department of CSE, IIT Bombay, Mumbai, India
[3]Department of CSE, IIT Madras, Chennai, India

Abstract

Information Extraction from Indian languages requires effective shallow parsing, especially identification of "meaningful" noun phrases. Particularly, for an agglutinative and free word order language like Marathi, this problem is quite challenging. We model this task of extracting noun phrases as a sequence labelling problem. A Distant Supervision framework is used to automatically create a large labelled data for training the sequence labelling model. The framework exploits a set of heuristic rules based on corpus statistics for the automatic labelling. Our approach puts together the benefits of heuristic rules, a large unlabelled corpus as well as supervised learning to model complex underlying characteristics of noun phrase occurrences. In comparison to a simple English-like chunking baseline and a publicly available Marathi Shallow Parser, our method demonstrates a better performance.

1 Introduction

One of the key steps of a Natural Language Processing (NLP) pipeline is chunking or shallow parsing. It allows the system to identify building blocks of a sentence namely phrases. Further, the identification of phrases is of importance to applications in information extraction, text summarization and event detection. In English, the task of chunking is relatively simple as compared to other steps of the NLP pipeline. Given the Parts-Of-Speech (POS) tags of the sentence tokens, it is a matter of using rules based on the POS tags (Abney, 1992) to extract the chunks with high confidence. State of the art papers on Noun Phrase (NP) chunking in English (Sun et al., 2008), (Shen and Sarkar, 2005), (McDonald et al., 2005) report results of more than 94% F-measure.

However, the scenario is different for many Indian languages. Their suffix agglutinative and free word order nature makes it challenging to identify the correct phrases. In this paper, we focus on the problem of identifying noun phrases from Marathi, a highly agglutinative Indian language. Marathi is spoken by more than 70 million people worldwide [1] and it exhibits a large web presence in terms of e-newspapers, Marathi Wikipedia, blogs, social network banter and much more. There are various motivations for the problem and the most important one being the lack of domain specific information extraction systems in Indian Languages. The problem also poses challenges in terms of NLP resource-poor nature of Indian languages and a complex suffix agglutination scheme in Marathi. Keeping these challenges in sight, we propose the use of distantly supervised approach for the task.

The task of identifying meaningful noun phrases in Marathi becomes challenging due to another fact that two different noun phrases can be written adjacently in a sentence. Such noun phrases are individually meaningful but their concatenation is not. Hence, it is important to correctly identify the boundaries of such noun phrases. We propose to model the task of identifying noun phrases as a sequence labelling task where the labelled data is generated automatically by using a set of heuristic rules. Moreover, these rules are not based on deep linguistic knowledge but are devised using corpus statistics.

In the next section, related work on Indian Language shallow parsing is presented. Section 3 describes a simple baseline for noun phrase identification for Marathi. Thereafter in Section 4, the distant supervision based sequence labelling approach is described in detail. This is followed by

[1] https://en.wikipedia.org/wiki/Marathi_language (accessed 11-AUG-2015)

a description of the corpus, experiments and evaluation results.

2 Related Work

This section describes relevant work in the field of shallow parsing and chunking in Indian languages. The work is presented based on a categorization into papers and tool sets.

Starting with the path setting paper on parsing of free word order languages (Bharati and Sangal, 1993), there have been multiple contributions to parsing of Indian Languages. A national symposium on modelling and shallow parsing of Indian languages held at IIT Bombay in April 2006 (MSPIL, 2006) brought together Indian NLP researchers to discuss on problems in Indian language NLP. Investigations in shallow parsing and morphological analysis in Bengali, Kannada, Telegu and Tamil were presented.

Also in 2006, a machine learning contest on POS tagging and chunking for Indian languages (NLPAI and IIIT-H, 2006) was organized which lead to release of POS tagged data (20K words) in Hindi, Bengali and Telugu and Chunk tagged data in Hindi. The participating systems employed various supervised machine learning methods to perform POS tagging and chunking for the three languages. The team from IIT Bombay (Dalal et al., 2006), trained a Maximum Entropy Markov Model (MEMM) from the training data to develop a chunker in Hindi and then evaluated it on test data. Their chunker was able to achieve an F1-measure of 82.4% on test data. In the entry by the team from IIT Madras (Awasthi et al., 2006), apart from a HMM based POS tagger, a chunker was developed by training a linear chain CRF using the MALLET toolkit. It achieved an overall F1-measure of 89.69% (with reference POS tags) and 79.58% (with the generated POS tags). Another system from Jadavpur University (Bandyopadhay et al., 2006) contributed a rule-based chunker for Bengali which comprised of a two stage approach - chunk boundary identification and chunk labelling. On an unannotated test set, the chunker reported an accuracy of 81.61%. The system from Microsoft Research India (Baskaran, 2006), comprised of a chunker for Hindi. It was developed by training a HMM and using probability models of certain contextual features. The system reported an F1-measure of 76% on the test set. The team from IIIT Hyderabad developed a chunker (Himashu and Anirudh, 2006) by training Conditional Random Fields (CRFs) and then evaluating on the test data. Their chunker performed at an F1-measure of 90.89%.

A more formal effort lead to the organization of the IJCAI workshop on Shallow Parsing for South Asian Languages and an associated contest (Bharati and Mannem, 2007), which brought out multiple contributions in POS tagging and shallow parsing of Hindi, Bengali and Telugu. Chunk data for Bengali and Telugu was also made available, however data of no other languages were introduced. In total, a set of 20,000 words for training, 5000 words for validation and 5000 words for testing were provided for all three languages. A listing of major contributions for the chunking task is presented as follows. A rule based system was proposed by Ekbal et al. (2007), where the linguistic rules worked well for Hindi (80.63% accurate) and Bengali (71.65% accurate). A Maximum Entropy Model based approach (Dandapat, 2007) worked well for Hindi (74.92% accurate) and Bengali (80.59% accurate). It used contextual POS tags as features than the context words. In another submission (Pattabhi et al., 2007), the technique involved a Transformation-Based Learning (TBL) for chunking and reported to have moderate results for Hindi and Bengali. The technique proposed in (Sastry et al., 2007) was tried at learning chunk pattern templates based on four chunking parameters. It used the CKY algorithm to get the best chunk sequence for a sentence. The results were moderate for all the three languages. Rao and Yarowsky (Rao and Yarowsky, 2007), observed that punctuations in the sentence act as roadblocks to learning clear syntactic patterns and hence they tried to drop them, leading to a rise in accuracy. However, they reported results on a different Naive Bayes based system. The system that came close second to the winner was by Agrawal (2007). It divided the chunking task into three stages - boundary labelling (BL), chunk boundary detection using labels from first stage and finally re-prediction of chunk labels. For Bengali and Telugu, their system performed almost at par with the the best system's accuracy. The system that performed the best on all three languages was proposed by PVS and Karthik (2007) which used HMMs for chunk boundary detection and CRFs for chunk labelling. They reached chunking accuracies of 82.74%, 80.97% and 80.95% for

Bengali, Hindi and Telugu respectively.

Apart from the two major exercises above, there has been a constant stream of work being published in the area. One of the older and primary effort was by Singh et al. (2005), where a HMM based chunk boundary identification system was developed by training it on a corpus of 1,50,000 words. The chunk label identification was rule based and the combined system was tested on a corpus of manually POS tagged 20,000 words leading to an accuracy of 91.7%. A more recent work on Malayalam shallow parsing (Nair and Peter, 2011) proposes a morpheme based augmented transition network for chunking and is reported to achieve good results on a small dataset used in the paper. Another important contribution by Gahlot et al. (2009) is an analysis paper on use of sequential learning algorithms for POS-tagging and shallow parsing in Hindi. The paper compares Maximum Entropy models, CRFs and SVMs on datasets of various sizes leading to conclusive arguments about the performance of the chosen systems. An important contribution is by Gune et al. (2010), where development of a Marathi shallow parser is explored using a sequence classifier with novel features from a rich morphological analyser. The resulting shallow parser shows a high accuracy of 97% on the moderate dataset (20K words) used in the paper.

On the tools front, there are various tool sets for shallow parsing in Indian languages which are available for public download. The most important and foremost ones being the shallow parsers provided by Language Technology Research Centre at IIIT-H (2015). The available shallow parsers are for languages - Hindi, Punjabi, Urdu, Bengali, Tamil, Telugu, Kannada, Malayalam and Marathi. We use this LTRC IIIT-H provided Marathi shallow parser as one of our baselines. Another set of shallow parsing tools are available from the CALTS, School of Humanities at University of Hyderabad (2015). They are focused on another set of languages namely Assamese, Bodo, Dogri, Gujarati, Hindi, Kashmiri, Konkani, Maithili, Manipuri, Nepali, Odia and Santali.

3 A Simple Baseline Approach

We define a noun phrase as a contiguous "meaningful" sequence of adjective (optional) followed by nouns. Here, we consider both types nouns: proper nouns and common nouns. Our definition also stresses on the "meaningfulness" of the sequence of nouns to be identified as a "valid" noun phrase.

Our observation is that extraction of such noun phrases is quite straight-forward in English and requires application of a simple regular expression on POS tagged text. In English, boundaries of such noun phrases are explicitly marked by prepositions, punctuations, determiners and verbs. Consider the sentence: `Ajay met Sachin Tendulkar in Mumbai.` Here, it can be observed that there are 3 "valid" noun phrases: `Ajay`, `Sachin Tendulkar` and `Mumbai`, which are perfectly separated by the verb `met` and the preposition `in`. There is no other way of writing this sentence in English such that any of the two noun phrases are written adjacently to each other. Hence, it is straight-forward in English to extract such noun phrases by writing a simple rule / regular expression. However, such a simple rule does not work for Indian languages like Marathi. Marathi has a free word order and is also highly agglutinative. In Marathi, the same sentence will be written as अजय सचिन तेंडुलकरला मुंबईमध्ये भेटला. Here, the first four words are nouns and hence the English-like phrase extraction rule will extract only one noun sequence (अजय सचिन तेंडुलकरला मुंबईमध्ये) which is not "meaningful".

We propose a simple baseline approach to extract "meaningful" noun phrases in Marathi which is a modification of the English-like phrase extraction rule. In Marathi, unlike English prepositions are not separate words but they are written as suffixes of the nouns. Hence, it is essential to remove suffixes attached to the words and write them as a separate token. This process of removing suffixes and identifying rootword for each word, is called as *Stemming*. After stemming, the above sentence becomes : अजय सचिन तेंडुलकर ला मुंबई मध्ये भेट ला. Now, if we extract all the consecutive nouns, we get 2 sequences : अजय सचिन तेंडुलकर and मुंबई. Here, the second sequence is a "valid" noun phrase but the first one is not.

Computationally, to use this baseline method it is necessary to apply stemming and POS tagging on a sentence to produce a sequence as follows:
अजय/NNP सचिन/NNP तेंडुलकर/NNP ला/SUF मुंबई/NNP मध्ये/SUF भेट/VM ला/SUF ./SYM
Then the following regular expression is applied for extraction of noun phrases.

(<word>/JJ)?(<word>/NNP?)*(<word>/NNP?)

This proposed baseline approach is quite efficient and effective. But unlike English, in Marathi all contiguous sequences of nouns (without any suffixes attached to these nouns) need not yield a "meaningful" noun phrase. This is because, in Marathi it is perfectly syntactical to have multiple consecutive noun phrases without any explicit (prepositions, verbs, punctuations etc.) or implicit (e.g. suffixes attached to words, change of POS from NN to NNP and vice versa) boundary markers.

4 Distantly supervised sequential labelling approach

"Distant supervision" is a learning paradigm in which a labelled training set is constructed automatically using some heuristics or rules. The resulting labelled data may have some noisy / incorrect labels but it is expected that majority of the automatically obtained labels are correct. Since it is possible to create a large labelled dataset (much larger than manually labelled data), majority correct labels will hopefully reduce the effect of a smaller number of noisy labels in the training set.

For any distant supervision based algorithm, there are two essential requirements - i) Large pool of unlabelled data and ii) Heuristic rules to obtain noisy labels. Distant supervision has been successfully used for the problem of Relation Extraction (Mintz et al., 2009). Semantic database like FreeBase (Bollacker et al., 2008) is used to get a list of entity pairs following any particular relation. Also, a large number of unlabelled sentences are used which can be easily obtained by crawling the Web. The labelling heuristic used here is: If two entities participate in a relation, any sentence that contains both of them might express that relation. For example, Freebase contains entity pair <M. Night Shyamalan, The Sixth Sense> for the relation ID /film/director/film, hence both of the following sentences are considered to be positive examples for that relation:

1. M. Night Shyamalan gained international recognition when he wrote and directed 1999's The Sixth Sense.
2. The Sixth Sense is a 1999 American supernatural thriller drama film written and directed by M. Night Shyamalan.

Though this assumption is expected to be true for majority of the sentences, it may introduce few noisy labels. For example, following sentence contains both the entities but does not express the desired relation.

1. The Sixth Sense, a supernatural thriller film, was written by M. Night Shyamalan.

4.1 Motivation

It is difficult to obtain labelled data where valid noun phrases are marked, because such a manual task is time consuming and effort intensive. In order to build a phrase identifier for Marathi, without spending manual efforts on creation of labelled data, we propose to use the learning paradigm of "Distant Supervision". Unlabelled data is easily available in this case, which is the first essential requirement of the distant supervision paradigm. We label sentences from a large unlabelled Marathi corpus with POS tags using a CRF-based POS tagger. We also check whether any suffixes are attached to the words and split such words into root word followed by suffixes as separate tokens [2]. The baseline method explained in the previous section, is then applied on all of these sentences to extract "candidate" noun phrases. As we have already described, this baseline method fails when two different noun phrases occur adjacent to each other. In order to devise some effective rules to create labelled data automatically, we take help of these candidate phrases. Based on corpus statistics (described in Section 4.2), we split some of the candidate phrases and keep other candidate phrases intact. In order to simplify the splitting decision, we assume that there will be at most one split point, i.e. any candidate phrase consists of at most two different consecutive noun phrases. After analysing a lot of Marathi sentences, we observed that this is a reasonable assumption to make because it is very rare to have more than 2 consecutive noun phrases.

4.2 Corpus Statistics

Various statistics of words and phrases are computed by using a large unlabelled corpus. These statistics are used in order to devise rules for distant supervision.

4.2.1 Phrase Counts:

We extract all the candidate phrases from the corpus using the first baseline method. For each candidate phrase, we note the number of times it oc-

[2] Marathi Stemmer by CFILT, IIT Bombay is used

curs in the corpus. Any candidate phrase which occurs more than 2 times in the corpus is likely to be a "valid" phrase.

4.2.2 Word Statistics:

Some useful statistics of words are computed using the list of "valid" noun phrases. For each word w, following counts are noted:

1. *Start_Count* : Number of times w occurs as a first word in a "valid" phrase with multiple words (E.g. बेकारी in the phrase बेकारी भत्ता)
2. *Unitary_Count* : Number of times w occurs as the only word in a "valid" phrase (E.g. पक्ष in the phrase पक्ष)
3. *Continuation_Count* : Number of times w is NOT the first or last word in a "valid" phrase with more than 2 words (E.g. क्रिकेट in the phrase भारतीय क्रिकेट संघ)
4. *End_Count* : Number of times w occurs as a last word in a "valid" phrase with multiple words (E.g. भत्ता in the phrase बेकारी भत्ता)

For each word, its most frequent category (out of *Start*, *End*, *Unitary* and *Continuation*) is stored in the structure *WordType*. Four different sets of words are defined : *Start_Words*, *Unitary_Words*, *Continuation_Words* and *End_Words*. If any word w occurs n times overall in the "valid" phrases and its *Start_Count* is at least $0.1 * n$, then the word w is added to the set *Start_Words*. The other three sets *Unitary_Words*, *Continuation_Words* and *End_Words* are similarly populated corresponding to the counts *Unitary_Count*, *Continuation_Count* and *End_Count*, respectively. One special set of words, *Unitary_Only_Words* is defined which contains all those words in *Unitary-Words* which are not present in any of the other 3 sets. Table 1 shows all these corpus statistics for some of the representative words.

4.3 Rules for Distant Supervision

With the help of the output of first baseline method and the corpus statistics, we devise heuristic rules for creating labelled data. For each candidate phrase p generated by the baseline method, following rules are applied sequentially.

4.3.1 Rule 1 (W1)

If p has only one word, then it is trivially correct. All the remaining rules are applied for only multi-word candidate phrases.

4.3.2 Rule 2 (AdjNoun)

If p has exactly two words such that the first word is an adjective and the second word in a noun, then p is a correct phrase.

4.3.3 Rule 3 (C3)

If p has corpus count of 3 or more, then it is very likely to be correct. But in order to make this rule more precise, some more constraints are applied to p. If the first word of p is in the set *Unitary_Words* but not in the set *Start_Words* or if the last word of p is in the set *Unitary_Words* but not in the set *Last_Words*, then then p is likely to be incorrect. Hence, excluding such phrases, this rule assumes all other candidate phrases with corpus count of at least 3 to be correct phrases.

4.3.4 Rule 4 (C2C2)

All the rules till now checked whether the candidate phrase as a whole is correct. This rule tries to estimate whether to split any candidate phrase into two consecutive meaningful phrases. If there are n words in a candidate phrase, then there are $(n-1)$ potential splits. Algorithm 1 *Split_Check* is used to determine whether any given split is valid or not. Algorithm 2 describes this rule in detail. In simple words, this rule splits a candidate phrase only if its sub-phrases are "valid" and each of the sub-phrase has been observed at least twice in the corpus.

Algorithm 1: *Split_Check* (checking validity of a split)

Data: Candidate phrase p, Split Index i, Corpus Statistics (as explained in Table 1)
Result: Whether splitting p at i is "valid"

$L_1 := i^{th}$ word of p ; `/* Last word of first sub-phrase */`
$F_2 := (i+1)^{st}$ word of p ; `/* First word of second sub-phrase */`
if L_1 *or* F_2 *are unseen words* **then return** *FALSE*;
if $L_1 \in$ Continuation_Words **then return** *FALSE*;
if $F_2 \in$ Continuation_Words **then return** *FALSE*;
if $i = 1$ **and** $L_1 \notin$ Unitary_Words **then return** *FALSE*;
if $i = len(p) - 1$ **and** $F_2 \notin$ Unitary_Words **then return** *FALSE*;
if $L_1 \in$ Start_Words **then return** *FALSE*;
if $F_2 \in$ End_Words **then return** *FALSE*;
return *TRUE*;

4.3.5 Rule 5

Similar to Rule 4, this rule also tries to estimate whether any candidate phrase can be split into two consecutive meaningful phrases. But unlike Rule

Word	*Corpus Count*	*Start Count*	*End Count*	*Unitary Count*	*Continuation Count*	*Member of Sets*	*Word Type*
लोकसभा *loksabhaa* Loksabha	640	**331**	7	**293**	9	*Start_Words* *Unitary_Words*	*Start*
गांधी *gandhee* Gandhi	567	16	**482**	48	21	*End_Words*	*End*
स्वयंसेवक *swayamsevak* volunteer	61	0	4	**16**	**41**	*End_Words* *Continuation_Words*	*Continuation*
तंत्रज्ञान *TanTradnyan* technology	302	8	**102**	**179**	13	*End_Words* *Unitary_Words*	*End*
निर्णय *nirNay* decision	2200	164	130	**1892**	14	*Unitary_Words* *Unitary_Only_Words*	*Unitary*

Table 1: Examples of various corpus statistics generated. Specific counts more than 10% of the total counts are shown in **bold**.

Algorithm 2: Rule 4 for splitting a candidate phrase using corpus counts

Data: Candidate phrase p, Corpus Statistics (as explained in Table 1)
Result: Two "valid" sub-phrases OR FALSE if no "valid" sub-phrases are found

$i := 1$;
while $i < length(p)$ **do**
  if Split_Check*(p,i)* = *FALSE* **then continue**;
  $i := i + 1$;
  $p_1 := p[1 : i]$; `/* First sub-phrase */`
  $p_2 := p[i + 1 : length(p)]$; `/* Second sub-phrase */`
  if $CourpusCount(p_1) < 2$ **then continue**;
  if $CourpusCount(p_2) < 2$ **then continue**;
  return (p_1, p_2);
end
return *FALSE*;

Algorithm 4: Rule 6 Unitary Split

Data: Candidate phrase p, Corpus Statistics (as explained in Table 1)
Result: Two "valid" sub-phrases OR FALSE if no "valid" sub-phrases are found

$p_1 := p[1]$; `/* First word of p */`
$p_2 := p[2 : length(p)]$; `/* Remaining words of p */`
if $p_1.POS = NN$ **and** $p_1 \in$ Unitary_Only_Words **then return** (p_1, p_2);
$p_2 := p[lenght(p)]$; `/* Last word of p */`
$p_1 := p[1 : lenght(p) - 1]$; `/* Remaining words of p */`
if $p_2.POS = NN$ **and** $p_2 \in$ Unitary_Only_Words **then return** (p_1, p_2);
return *FALSE*;

4 which uses corpus counts, this rule uses properties of the words at split boundary. Algorithm 3 explains this rule in detail.

4.3.6 Rule 6 (UnitarySplit)

Like rules 4 and 5, this rule also checks whether a candidate phrase can be split. This rule handles the specific case of a single common noun (NOT proper noun) adjacent to other meaningful phrase. It does not check the validity of a split by using the algorithm *Split_Check* (Algorithm 1) but uses a stricter check to validate "Unitary" nature of such single common nouns. The detailed explanation is provided in the Algorithm 4.

4.3.7 Rule 7 (W2*)

This is the default rule applied on those candidate phrases for which none of the earlier rule is satisfied. In other words, rules 1 to 3 are not able to identify these phrase as "valid" phrases as a whole. Also, rules 4 to 6 are not able to identify a "valid" split to produce two consecutive meaningful phrases. This rule assumes that all such phrases are "valid" and keeps them intact without any split.

4.4 Estimated Accuracy of Rules

All the rules explained in the Section 4.3 are applied on a large unlabelled corpus (approximately 200,000 sentences) and labelled data is automatically produced. These automatically labelled sentences are further used to train a sequence classifier (CRF in our case) so that it can be used to extract proper noun phrases from any unseen sentence. Automatically obtained phrase labels may contain some noise. In order to get an estimate of accuracy of the labels, for each rule we collected random sample of 100 candidate phrases. These samples were manually verified to get an estimate of accuracy of each rule which are shown in the

Algorithm 3: Rule 5 for splitting a candidate phrase using word properties

Data: Candidate phrase p, Corpus Statistics (as explained in Table 1)
Result: Two "valid" sub-phrases OR FALSE if no "valid" sub-phrases are found

$i := 1$;
while $i < length(p)$ **do**
  if Split_Check*(p,i)* = *FALSE* **then continue**;
  $i := i + 1$;
  $p_1 := p[1 : i]$; /* First sub-phrase */
  $p_2 := p[i + 1 : length(p)]$; /* Second sub-phrase */
  $L_1 := i^{th}$ word of p ; /* Last word of first sub-phrase */
  $F_2 := (i + 1)^{st}$ word of p ; /* First word of second sub-phrase */
  if $length(p_1) = 1$ **and** $length(p_2) = 1$ **then**
    if $L_1 \in$ Unitary_Words **and** $F_2 \in$ Unitary_Words **and** $L_1.POS \neq F_2.POS$ **then return** (p_1, p_2);
    if $WordType[L_1] = Unitary$ **and** $WordType[F_2] = Unitary$ **then return** (p_1, p_2)
  end
  if $length(p_1) = 1$ **and** $length(p_2) > 1$ **then**
    if $L_1 \in$ Unitary_Words **and** $F_2 \in$ Start_Words **and** $L_1.POS \neq F_2.POS$ **then return** (p_1, p_2);
    if $WordType[L_1] = Unitary$ **and** $WordType[F_2] = Start$ **then return** (p_1, p_2);
  end
  if $length(p_1) > 1$ **and** $length(p_2) = 1$ **then**
    if $L_1 \in$ End_Words **and** $F_2 \in$ Unitary_Words **and** $L_1.POS \neq F_2.POS$ **then return** (p_1, p_2);
    if $WordType[L_1] = End$ **and** $WordType[F_2] = Unitary$ **then return** (p_1, p_2);
  end
  if $length(p_1) > 1$ **and** $length(p_2) > 1$ **then**
    if $L_1 \in$ End_Words **and** $F_2 \in$ Start_Words **and** $L_1.POS \neq F_2.POS$ **then return** (p_1, p_2);
    if $WordType[L_1] = End$ **and** $WordType[F_2] = Start$ **then return** (p_1, p_2);
  end
end
return *FALSE*;

Table 2. Higher the number of phases labelled by a rule, higher is its *Support* and lower the number of estimated errors, higher is its *Confidence*.

Rule	#Candidate Phrases Labelled	#Errors in Random Sample of 100
Rule 1 (W1)	632735	0
Rule 2 (AdjNoun)	56133	0
Rule 3 (C3)	56401	4
Rule 4 (C2C2)	22883	16
Rule 5 (ValidSplit)	2158	32
Rule 6 (UnitarySplit)	23948	51
Rule 7 (W2*)	102387	36

Table 2: Estimated Accuracy of the Rules used for Distant Supervision

Here, it is to be noted that "Error" is rule specific. For a non-splitting rule like rule 3, an error will occur when it keeps an incorrect candidate phrase intact which should have been split. However, for a splitting rule like rule 4, an error will occur when it splits a "valid" phrase which should not have been split. It can be observed that rule 6 (UnitarySplit) is the least accurate rule, but we still use it because we believe that the noise introduced by it can be overcome by the sequence classifier because of other better performing rules having more coverage.

4.5 Sequence Labelling

The intuition behind learning a sequence labelling model is that such a statistical model can implicitly learn several more complex rules to identify meaningful noun phrases. We use Conditional Random Fields (CRF) (Lafferty et al., 2001) for sequence labelling. In this section, we describe how the labelled data is automatically created for training CRF model and what are the various features used by the CRF model.

4.5.1 Creation of Training Data for CRF

Our unlabelled corpus contains around 200,000 sentences. Approximately 55,000 sentences have at least two candidate phrases labelled by any of the rules from rule 1 to rule 6. These sentences are used for training sequence labelling model using CRF. Table 3 shows some examples of candidate phrases labelled by the rules. We are using the BIO labelling scheme which assigns one of the three different labels to each sequence element (word or suffix in our case) as follows:
B: First word in a phrase is labelled as **B**.
I: All subsequent words except the first word in a phrase are labelled as **I**.
O: All other words or suffixes which do not belong to any phrase are labelled as **O**.

Rule	Labelled Phrases
Rule 1 (W1)	संधी/B बातमी/B
Rule 2 (AdjNoun)	अपेक्षित/B बदल/I मराठी/B माणूस/I
Rule 3 (C3)	परिवहन/B विभाग/I भारतीय/B क्रिकेट/I संघ/I
Rule 4 (C2C2)	आरोप/B भाजप/B तपशील/B कामगार/B आयुक्त/I
Rule 5 (ValidSplit)	कार्यभार/B प्रभारी/B अधिकारी/I मालक/B राहुल/B लिमये/I
Rule 6 (UnitarySplit)	प्रयत्न/B फायर/B ब्रिगेड/I घोषणा/B सरकार/B
Rule 7 (W2*)	शास्त्रशुद्ध/B प्रशिक्षण/I विनायक/B कम्प्युटर्स/I

Table 3: Examples of automatically labelled candidate phrases using distant supervision rules

4.5.2 Features used by the CRF classifier

In general, there are two types of features used in a CRF model : unigram and bigram. Unigram features are combination of some property of the sequence w.r.t. the *current* token and the *current* label. Bigram features are combination of some property of the sequence w.r.t. the *current* token, *current* label and *previous* label. For every i^{th} token (word or suffix) in a sequence, following classes of unigram features are generated.

1. Lexical Features: Word or suffix at the positions $i, (i-1)$ and $(i+1)$. If current word belongs to any candidate phrase, then the words preceding and succeeding that candidate phrase are also considered as features. If the current word does not belong to any candidate phrase, then values of these features are "NA".

2. POS Tag Features: POS tags of words at the positions $i, (i-1), (i-2), (i+1)$ and $(i+2)$. If current word belongs to any candidate phrase, then the POS tags of the words preceding and succeeding that candidate phrase are also considered as features. If the current word does not belong to any candidate phrase, then values of these features are "NA".

Similary, for every i^{th} token (word or suffix) in a sequence, following classes of bigram features are generated.

1. Edge Features: Combination of labels at positions i and $(i-1)$.

2. POS Tag Edge Features: Combination of POS tag at position i and labels at positions i and $(i-1)$.

5 Experiments and Evaluation

5.1 In-house POS Tagger

We developed an in-house POS tagger by training a CRF on the NLTK(Bird et al., 2009) Indian languages POS-tagged corpus for Marathi. Features like prefixes, suffixes, rootwords and dictionary categories of the tokens were used while training. The POS tagger was evaluated based on a 80-20 train-test split of the NLTK data and the accuracy of 91.08% was obtained.

5.2 Corpus

As described earlier, distant supervision allows creation of large amount of training data through heuristics. However, the first requirement is a large corpus of Marathi text on which such heuristics can be applied. We considered the Marathi FIRE Corpus (Palchowdhury et al., 2013) which has crawled archives of a Marathi Newspaper (Maharashtra Times [3]) for 4 years starting from 2004 to 2007. We used all the articles of year 2004 for the task. After a trivial preprocessing we were able to extract all the text sentences from the files for compiling the corpus. The corpus comprises of about 200,000 sentences.

5.3 Evaluation using Test Dataset

We created a test dataset of 100 sentences and manually identified all the "valid" noun phrases in them. Apart from the baseline approach using English-like phrase extraction rule, we also consider two other baselines: i) the LTRC IIIT-H provided Marathi Shallow Parser and ii) Applying Heuristic rules (defined in the section 4.3) directly on the sentences of test dataset. We evaluated all the baseline methods and our distantly supervised CRF by applying them on this test dataset. For each method, the gold-standard set of noun phrases was used to evaluate the set of noun phrases extracted by that method by computing the following:

True Positives (TP) : Number of extracted phrases which are also present in the set of gold-standard phrases.

False Positives (FP) : Number of extracted phrases which are not present in the set of gold-standard phrases.

False Negatives (FN) : Number of gold-standard phrases which are not extracted.

[3] http://maharashtratimes.indiatimes.com/

Method	P (%)	R (%)	F1 (%)
IIITH Shallow Parser	83.38	88.68	85.95
Baseline 1 (English-like rule)	88.89	85.07	86.94
Baseline 2 (Heuristic Rules)	87.43	89.72	88.56
Distantly supervised CRF	88.31	89.14	**88.72**

Table 4: Comparative performance of various methods on a test dataset of 100 labelled sentences

The overall performance of any method is then measured in terms of *Precision* (P), *Recall* (R) and *F-measure* (F) as follows:
$P = \frac{TP}{(TP+FP)}, R = \frac{TP}{(TP+FN)}, F = \frac{2PR}{(P+R)}$

Table 4 shows the comparative performance of all the methods and it can be observed that our method outperforms all the baselines. Also, the performance improvement of distantly supervised CRF over the heuristic rules is not very significant. We plan to carry out more detailed analysis of this phenomenon in future, by experimenting with additional features in the CRF model.

5.4 Analysis

After analyzing the error cases, we found that one of the major reasons for the errors was the lack of sufficient corpus statistics for some of the words (especially proper nouns). Consider the following sentence from our test dataset : गेल्या आठवड्यात चंदाबाबू लायन्स क्लबच्या एका मेळाव्यास संबोधित करण्यासाठी अहमदाबादेसही जाऊन आले (`Chandababu had been to Ahmedabad last week for addressing a gathering of the Lions Club.`) Here, all the methods (IIIT-H Shallow parser, baseline methods as well as our method using CRF) incorrectly identify the phrase चंदाबाबू लायन्स क्लब as a single "meaningful" noun phrase. Ideally, चंदाबाबू and लायन्स क्लब are two separate phrases. Here, both the words चंदाबाबू and लायन्स are proper nouns and they occur rarely in the corpus, resulting in unreliable corpus statistics for these words.

However, most of the common nouns have significant presence in the corpus producing reliable corpus statistics for these words. As our method is heavily dependent on the corpus statistics, such cases involving frequent words are handled correctly. Consider following sentence from our test dataset: देशभरातून लाखो भाविक दर्शनासाठी तिथे जात असतात (`From across the country, lacs of devotees keep going there for the auspicious sight`). Here, the noun phrases extracted by the IIIT-H Shallow Parser are देशभरातून and लाखो भाविक दर्शनासाठी whereas our method correctly identifies 3 noun phrases : देशभर, लाखो भाविक and दर्शन. Here, both the words लाखो भाविक and दर्शन have reliable corpus statistics and hence our method correctly splits the candidate phrase लाखो भाविक दर्शन producing "meaningful" noun phrases.

6 Conclusion and Future Work

We highlighted an important problem of extraction of "meaningful" noun phrases from Marathi sentences. We propose a distant supervision based sequence labelling approach for addressing this problem. A novel set of rules based on corpus statistics are devised for automatically creating a large labelled data. This data is used for training a CRF model. Most other approaches to chunking for Indian languages are supervised and need large corpus of labelled training data. The main advantage of our work is that it does not need manually created labelled training data, and hence can be used for resource-scarce Indian languages. Our approach not only reduces the efforts for creation of labelled data but also demonstrate better accuracy than the existing approaches.

As our rules for distant supervision are based on corpus statistics and not on deep linguistic knowledge, they can be easily ported to other Indian languages. In future, we would like to take our work further on these lines. We also plan to extend our framework to a full-scale chunking tool for Marathi, not just noun phrases. Additionally based on this work, we plan to build a generic Information Extraction engine for Marathi and later for other Indian languages.

References

Steven P Abney. 1992. *Parsing by chunks*. Springer.

Himanshu Agrawal. 2007. POS tagging and chunking for Indian languages. In *IJCAI Workshop On Shallow Parsing for South Asian Languages (SPSAL)*.

Pranjal Awasthi, Delip Rao, and Balaraman Ravindran. 2006. Part of speech tagging and chunking with HMM and CRF. In *NLPAI Machine Learning Contest*.

Sivaji Bandyopadhay, Asif Ekbal, and Debasish Halder. 2006. HMM based POS tagger and rule-based chunker for bengali. In *Sixth International Conference on Advances In Pattern Recognition*, pages 384–390. World Scientific.

Sankaran Baskaran. 2006. Hindi POS tagging and chunking. In *NLPAI Machine Learning Contest*.

Akshar Bharati and Prashanth R Mannem. 2007. Introduction to shallow parsing contest on South Asian languages. In *IJCAI Workshop On Shallow Parsing for South Asian Languages (SPSAL)*, pages 1–8. Citeseer.

Akshar Bharati and Rajeev Sangal. 1993. Parsing free word order languages in the paninian framework. In *ACL*, pages 105–111. ACL.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python*. " O'Reilly Media, Inc.".

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *ACM SIGMOD international conference on Management of data*, pages 1247–1250. ACM.

Aniket Dalal, Kumar Nagaraj, Uma Sawant, and Sandeep Shelke. 2006. Hindi part-of-speech tagging and chunking: A maximum entropy approach. In *NLPAI Machine Learning Contest*.

Sandipan Dandapat. 2007. Part of specch tagging and chunking with maximum entropy model. In *IJCAI Workshop On Shallow Parsing for South Asian Languages (SPSAL)*, pages 29–32.

Asif Ekbal, S Mondal, and Sivaji Bandyopadhyay. 2007. POS tagging using HMM and rule-based chunking. In *IJCAI Workshop On Shallow Parsing for South Asian Languages (SPSAL)*, pages 25–28.

Himanshu Gahlot, Awaghad Ashish Krishnarao, and DS Kushwaha. 2009. Shallow parsing for hindi-an extensive analysis of sequential learning algorithms using a large annotated corpus. In *Advance Computing Conference, 2009. IACC 2009. IEEE International*, pages 1158–1163. IEEE.

Harshada Gune, Mugdha Bapat, Mitesh M Khapra, and Pushpak Bhattacharyya. 2010. Verbs are where all the action lies: experiences of shallow parsing of a morphologically rich language. In *COLING : Posters*, pages 347–355. ACL.

Agarwal Himashu and Amni Anirudh. 2006. Part of Speech Tagging and Chunking with Conditional Random Fields. In *NLPAI Machine Learning Contest*.

LTRC IIIT-H. 2015. Language technology research centre, IIIT-H. [Online; accessed 20-August-2015].

John Lafferty, Andrew McCallum, and Fernando CN Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*.

Ryan McDonald, Koby Crammer, and Fernando Pereira. 2005. Flexible text segmentation with structured multilabel classification. In *HLT-EMNLP*, pages 987–994. ACL.

Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *ACL-IJCNLP, Volume 2*, pages 1003–1011. ACL.

MSPIL. 2006. Modelling and shallow parsing of Indian languages. [Online; accessed 20-August-2015].

Latha R Nair and S David Peter. 2011. Shallow parser for malayalam language using finite state cascades. In *Biomedical Engineering and Informatics (BMEI)*, volume 3, pages 1264–1267. IEEE.

NLPAI and IIIT-H. 2006. NLPAI contest on POS tagging and shallow parsing of Indian languages. [Online; accessed 20-August-2015].

University of Hyderabad. 2015. Calts lab, school of humanities, university of hyderabad. [Online; accessed 20-August-2015].

Sauparna Palchowdhury, Prasenjit Majumder, Dipasree Pal, Ayan Bandyopadhyay, and Mandar Mitra. 2013. Overview of FIRE 2011. In *Multilingual Information Access in South Asian Languages*, pages 1–12. Springer.

RK Pattabhi, T Rao, R Vijay Sundar Ram, R Vijayakrishna, and L Sobha. 2007. A text chunker and hybrid POS tagger for Indian languages. In *IJCAI Workshop On Shallow Parsing for South Asian Languages (SPSAL)*.

Avinesh PVS and G Karthik. 2007. Part-of-speech tagging and chunking using conditional random fields and transformation based learning. *Shallow Parsing for South Asian Languages*, 21.

Delip Rao and David Yarowsky. 2007. Part of speech tagging and shallow parsing of Indian languages. *Shallow Parsing for South Asian Languages*, page 17.

GM Ravi Sastry, Sourish Chaudhuri, and P Nagender Reddy. 2007. An HMM based part-of-speech tagger and statistical chunker for 3 Indian languages. In *IJCAI Workshop On Shallow Parsing for South Asian Languages (SPSAL)*.

Hong Shen and Anoop Sarkar. 2005. *Voting between multiple data representations for text chunking*. Springer.

Akshay Singh, Sushma Bendre, and Rajeev Sangal. 2005. HMM based chunker for Hindi. In *IJCNLP*.

Xu Sun, Louis-Philippe Morency, Daisuke Okanohara, and Jun'ichi Tsujii. 2008. Modeling latent-dynamic in shallow parsing: a latent conditional model with improved inference. In *COLING-Volume 1*, pages 841–848. ACL.