A Hybrid Approach for Bracketing Noun Sequence

Arpita Batra
LTRC, IIIT-Hyderabad
Hyderabad, India
arpita.batra@research.iiit.ac.in

Soma Paul
LTRC, IIIT-Hyderabad
Hyderabad, India
soma@iiit.ac.in

Abstract

For a resource poor language like Hindi, it becomes very difficult to bracket a noun sequence using approaches which are only based on corpus or lexical database. For semantic knowledge, power of both type of resources is needed to be combined. Therefore, affinity in between two nouns is preferred to be measured using backoff association which is the combination of lexical and conceptual association. Also, syntax is important for this task. But syntactic rules do not work for the compound nouns which is a special case of noun sequences and it may also occur as the sub-sequence. Using hybrid approach, accuracy of 86.33% has been obtained.

We have explored different variations like smoothing, frequency of synonyms and similar words for lexical association. And for conceptual association, different possible noun classes have been used for experiments. Authors have their own way of writing. Sometimes, two nouns can be written together as a single word or dash can be inserted in between the two. This helps in knowing that the two nouns have the tendency to be grouped together and hence this feature has been incorporated for the methods based on conceptual association.

1 Introduction

Complex noun sequence (NS) of Hindi was first discussed in Batra et al. (2014). We define NS as a sequence of nouns in which the noun constituents may or may not be separated by the genitives - "kA", "kI" or "ke".[1] When nouns occur without any intervening postpositions, such special cases are called compound nouns (CN). Batra et al. (2014) have noticed that compound nouns have a tendency to be grouped first within a complex NS. They have called this concept, *local grouping*. This grouping takes place due to indeclinable nature of all the nouns except the last one in a CN. This can be formalized using following context-free grammar:[2]

$G = \{V, \sum, R, NS\}$
$V = \{NS, CN, genitive, noun\}$
$\sum$ = set of common nouns $\cup$ set of genitives

And rules R for this grammar are given by:
NS $\rightarrow$ NS genitive NS | CN | noun
CN $\rightarrow$ CN CN | noun CN | CN noun | noun noun
genitive $\rightarrow$ kA | kI | ke
noun $\rightarrow$ daravAja | kursI | kamarA | ...
"door" "chair" "room"

Further, genitives and the head noun of the

[1] These allomorphic forms vary with gender, number and case values. Gender can have value - male(m) or female(f), number can have value - singular(s) or plural(p) and case can have - direct(d) or oblique(o) value. Morphological properties of genitives are:
kA - msd
kI - fsd or fso or fpd or fpo
ke - mso or mpd or mpo

[2] Context-free grammar has set of recursive rules which are used for generating strings from the non-terminals or the terminals (alphabets). It is represented using four tuples $\{V, \sum, R, S\}$
V is the set of non-terminals
$\sum$ is the set of terminals
R has the set of production rules from V to $(V \cup \sum)*$
S is the start symbol

D S Sharma, R Sangal and E Sherly. Proc. of the 12th Intl. Conference on Natural Language Processing, pages 276–284, Trivandrum, India. December 2015.

sequence occurring just after the genitive should be in agreement.[3] Therefore, set V and rules can be expanded as shown below:

$$V = \{NS_{\text{msd}}, NS_{\text{mso}}, NS_{\text{mp}}, NS_{\text{f}}, CN_{\text{msd}}, CN_{\text{mso}}, CN_{\text{mp}}, CN_{\text{f}}, genitive_{\text{msd}}, genitive_{\text{mso}}, genitive_{\text{mp}}, genitive_{\text{f}}, noun_{\text{msd}}, noun_{\text{mso}}, noun_{\text{mp}}, noun_{\text{f}}\}$$

$\text{NS} \rightarrow \text{NS}_{\text{msd}} \mid \text{NS}_{\text{mso}} \mid \text{NS}_{\text{mp}} \mid \text{NS}_{\text{f}}$
$\text{CN} \rightarrow \text{CN}_{\text{msd}} \mid \text{CN}_{\text{mso}} \mid \text{CN}_{\text{mp}} \mid \text{CN}_{\text{f}}$
$\text{genitive} \rightarrow \text{genitive}_{\text{msd}} \mid \text{genitive}_{\text{mso}} \mid \text{genitive}_{\text{mp}} \mid \text{genitive}_{\text{f}}$
$\text{noun} \rightarrow \text{noun}_{\text{msd}} \mid \text{noun}_{\text{mso}} \mid \text{noun}_{\text{mp}} \mid \text{noun}_{\text{f}}$

$\text{NS}_{\text{f}} \rightarrow \text{NS genitive}_{\text{f}}\ \text{NS}_{\text{f}} \mid \text{CN}_{\text{f}} \mid \text{noun}_{\text{f}}$
$\text{CN}_{\text{f}} \rightarrow \text{CN CN}_{\text{f}} \mid \text{noun CN}_{\text{f}} \mid \text{CN noun}_{\text{f}} \mid \text{noun noun}_{\text{f}}$
$\text{genitive}_{\text{f}} \rightarrow \text{kI}$
$\text{noun}_{\text{f}} \rightarrow \text{kursI} \mid \text{kursiyAZ} \mid \text{kursiyoM} \mid ...$
"chair" "chairs"[4] "chairs"[5]

Similarly, rules can be expanded for NS_{msd}, NS_{mso} and NS_{mp}.

There exists an implicit relationship in between the noun constituents of a sequence. This semantic relation can not be captured using context-free grammar.

Parsing of complex NS is a very significant task for its correct interpretation. Parsing is the task of recognizing input string and assigning a structure to it (Jurafsky and Martin, 2000). Semantic parsing is important for understanding the meaning of a sequence. For this, nouns which have an implicit relationship in between them should be identified and then semantic role can be assigned to it. Bracketing helps in knowing the sub-sequence which are possible to be grouped together. It is difficult to add all the sequences in a dictionary because of the many possible combinations (Yosiyuki et al., 1994). And hence, to retrieve the meaning of a sequence, we cannot take dictionary as the reference. Some algorithm is needed for such type of work. Here, we have used approaches based on semantic and syntactic knowledge which are discussed later. Example for some noun sequences with bracketing is shown below:

1. *((vidhAnasabhA chunAva) prachAra)*
   "assembly" "election" "propaganda"

2. *(sarakAra kI (gaThana nIti))*
   "government" *gen*[6]"formation" "policy"

3. *((gAoM ke nAgarikoM) kI madada)*
   "village" *gen* "citizens" *gen* "help"

The NS in example 3 can have two readings in two contexts as illustrated in the following examples:

(a) *gAoM ke nAgarikoM kI madada karo*
"village" gen "citizens" gen "help" "do"
"help the citizens of village"

(b) *gAoM ke nAgarikoM kI madada dvArA ye kAma huA*
"village" gen "citizens" gen "help" "by" "this" "work" "happened"
"This work happened by the help of village citizens"

The two interpretation can be represented using the following logical expressions:

(a) location(gAoM, nAgarikoM) && beneficiary(nAgarikOM, madada)

(b) location (gAoM, nAgarikoM) && agent (nAgarikOM, madada)

The meaning of the sequence can be ambiguous because of internal structure of the NS. Example:
gAon ke kisAnoM ke kheta
"village" *gen* "farmer" *gen* "farm"

It can be bracketed in two possible ways: (a) *((gAon ke kisAnoM) ke kheta)* and (b) *(gAon ke (kisAnoM ke kheta))*. Here, *gAon* refers to physical space which we can tag as village#1. The expression (a) conveys that farmers of the village#1 own farms and those farms are not necessarily located in the village#1, while in (b), farms are owned by some farmers which are located in village#1, but it is not necessary that farmers live in village#1. All the above cases of ambiguities can be resolved using contextual information. Legitimate bracketing will help in

[3] Agreement should be with gender, number and case
[4] plural(p) and direct(d)
[5] plural(p) and oblique(o)
[6] genitive

correct interpretation of NS.

In all the above examples, there are *n-1* pairs of modifier and modified for a sequence containing *n* number of nouns. In example 3, "gAoM" and "nAgarikoM", "nAgarikoM" and "madada" are the two pairs. This can be represented using a tree in which parent nodes are modified and all the children nodes are modifiers. This type of structure is known as *modificational structure*. Knowing about modifiers and modifieds is the part of semantic parsing. And, every binary parse tree has an equivalent modificational structure (Lauer and Dras, 1994). Therefore, the task of bracketing becomes important for semantic parsing. And it also becomes important for information extraction and question answering. This task can also be used for inter-chunk and intra-chunk dependency parsing.

As we know, affinity of two nouns is judged semantically. Therefore, in Section 3, we have proposed algorithms which parse the sequence by judging the combination of nouns semantically. In Section 4, we have shown how agreement and grouping of compound noun can help in improving further accuracy, if applied first.

2 Related Work

We know that compound noun is a special case of noun sequences and we have included them in our study. In literature, many methods have been tested and proposed for parsing complex compound nouns in English. Some of the related works are discussed here in chronological order.

Grouping of two constituents in a compound noun depends on the affinity in between them. And this is determined semantically (Marcus, 1980; Lauer, 1994). Marcus (1980) has proposed a solution in which the better noun-noun pair (n_1n_2 or n_2n_3) in a compound noun $n_1n_2n_3$ is found. If either of the sub-constituent n_1n_2 or n_2n_3 is unacceptable, then the other parse is chosen. If both are acceptable, then the one with higher preference is chosen. The method of deciding preference order is not mentioned in the work. Liberman and Sproat (1992) have used mutual information for deciding the preference while Pustejovsky et al. (1993) have compared the bigram frequency directly.

Marcus (1980) has proposed a method for parsing compound noun with more than three noun constituents. A buffer window of three constituents is taken. Initially, two components are grouped together from buffer. After this grouping, buffers are re-filled with the combined components, the component which has not been combined and the next component in a compound. The procedure of combining and filling the buffer is repeated till there exists the possibility of filling all the three buffers. This approach is a greedy approach and will fail for the compounds in which three leftmost nouns act only as the modifiers and not as the modified. Example of such parse structure are $(n_1(n_2(n_3n_4)))$, $((n_1(n_2(n_3n_4)))n_5)$ etc.

In previous methods, parsing is done using "lexical association". In such methods, grouping of nouns is resolved with the help of judging lexical preferences (Hindle and Rooth, 1993). Analysis which is based on lexical relationships faces the problem of data sparsity (Resnik, 1992; Resnik and Hearst, 1993). Word class information can also be used to measure the association. This type of association is termed as conceptual association (Resnik and Hearst, 1993). Lauer (1994) has used this concept for bracketing compound noun. This also helps in reducing the size of training data and is based on the assumption that all the nouns within a category behave in a similar manner (Lauer and Dras, 1994).

For training the data, Lauer (1994) has measured conceptual association using mutual information. A compound $n_1n_2n_3$ can be bracketed in two possible ways: $((n_1n_2)n_3)$ and $(n_1(n_2n_3))$. In first case, n_1 is modifying n_2 and n_2 is modifying n_3. And in second case, n_2 is modifying n_3 and n_1 is modifying n_3. This type of structure analysis is referred as *modificational structure* (Lauer and Dras, 1994). Models based on this concept were called dependency models and the models mentioned above were termed as adjacency models. Previous models were known as adjacency because the sub-sequences for which association is measured are adjacent to each other. For compound nouns with more than three nouns, they have proposed to multiply the

conceptual associations. Lauer (1995) has shown that the dependency models perform better than the adjacency models. They have used similar algorithm with lexical association and have found that the method which uses conceptual association performs better. While, Lapata and Keller (2005) have shown that lexical association measure performs equivalent to the conceptual association measure when frequency for a sub-sequence is obtained from web.

Nakov and Hearst (2005) have also extracted lexical statistics from web search engine and have found that chi-square performs better than the measure which is similar to mutual information. For improving further accuracy, they have used some features like dash, possessive marker, capitalisation. They have searched the compound noun with these cues and have used this result to improve the score of both left and right bracketing. If the number of cues which support left bracketing are greater than the number of cues supporting right bracketing, then the result is left bracketed parse. Otherwise the result is right bracketed parse.

Girju et al. (2005) have used a C5.0 decision tree to determine the parse of noun compounds with three nouns. Three top semantic classes of all nouns were used as the feature.

Kulkarni and Kumar (2011) have used conditional probability to determine the constituency parse. In Sankrit, compounds are not separated by spaces and therefore, compatibity is decided after segments are obtained using a segmenter.

Kavuluru and Harris (2012) have used both non-greedy (global) and greedy based approach for bracketing compound nouns having four constituents ($n_1n_2n_3n_4$). For greedy approach, they have directly compared n-gram frequency as done by Pustejovsky et al. (1993). To determine the parse using non-greedy approach, cohesion value for all possible parse trees is calulated and the one with the highest value is chosen as the correct parse. Cohesion means togetherness and its value is obtained by calculating sum of jaccard index for each non-leaf node of a tree. Their approach uses adjacent sub-sequences for measuring association.

Not much work has been done in the area of parsing complex noun sequences. Sharma (2011) has used six syntactic and four semantic rules for parsing recursive genitive construction which is also the special case of noun sequences. Syntactic rules fail for the cases when the two genitive markers in the construction are same. For such cases, semantic rules are applied. List of words are classified using time, direction and measurement information which helps in applying semantic rules. And Batra et al. (2014) have used four approaches: adjacency greedy, adjacency global, dependency greedy and dependency global for constituency parsing. They have shown that dependency global performs the best. To the best of my knowledge, no other work has been done for noun sequences.

3 Parsing using Semantic Knowledge

Parsing the sequence requires world knowledge. It is not necessary that all information can be obtained from a corpus. Corpus can also be domain based and can be of limited size. Then for such cases, it becomes very difficult to find the parse using only lexical association. Generally, methods which use conceptual association performs better. For a resource poor language, conceptual association is also not sufficient because of the less coverage of words. Therefore, we propose a method which is the combination of lexical and conceptual association. First, we have covered the methods which depends on lexical association and then conceptual association based method has been explored.

Hindi is poor in resources. It is very difficult to find the sub-sequence formed by noun constituents with more than two nouns. Therefore, good accuracy can not be obtained when lexical association is measured between the whole constituents as proposed by Kavuluru and Harris (2012). Batra et al. (2014) calls this as adjacency global approach and had proven that dependency global approach performs better than this. This is based on modificational structure and can be obtained from binary parse tree by converting head noun of the left child into a modifier which modifies head noun of the right child. Cohesion value (CV) is measured by summing the association value (AV) for each node which have two children. For finding the best

parse tree, cohesion value for all possible trees is calculated. The tree which has highest value is the result of bracketing.

$$CV(tree) = \sum_{\substack{n \in node \\ n \neq leaf\ node}} AV(H(lc(n)), H(rc(n))) \quad (1)$$

Association between head (H) of left and right child (*lc* and *rc*) of a non-leaf node 'n' can be found using lexical association or conceptual association. Lexical association uses frequency of noun constituents while conceptual association uses frequency of noun constituent's class. Association can be found using various measures. Some of the prevailing measures are pointwise mutual information, chi-square, jaccard index. Yang and Pedersen (1997) had shown that chi-square performs better than mutual information. This also has been shown by Nakov and Hearst (2005). Formula for jaccard index(ji), normalised pointwise mutual information(npmi) and chi-square(cs) are:

$$AV_{\text{ji}} = \frac{A}{A + B + C} \quad (2)$$

$$AV_{\text{ji}} = \frac{A}{freq(n_1) + freq(n_2) - A} \quad (3)$$

$$AV_{\text{npmi}} = \frac{\ln \frac{p(n_1,n_2)}{p(n_1)*p(n_2)}}{-\ln p(n_1, n_2)} \quad (4)$$

$$AV_{\text{cs}} = \frac{N(AD - BC)^2}{(A + C)(B + D)(A + B)(C + D)} \quad (5)$$

where,
A=freq($n_1 n_2$)
B=freq($n_1 \overline{n}_2$) = freq(n_1) - A
C=freq($\overline{n}_1 n_2$) = freq(n_2) - A
D=freq($\overline{n}_1 \overline{n}_2$)
N=A+B+C+D

$p(n_1, n_2)$ = A/N
$p(n_1)$ = freq(n_1)/N
$p(n_2)$ = freq(n_2)/N

It is found that normalized pointwise mutual information is working better for lexical association and jaccard index is performing better for conceptual association.

It is also not easy to find the bigrams in corpus. The count can be zero due to two reasons. First, two nouns can be combined but is unavailable. Another possibility is that the two nouns cannot be combined. If count of n_1 or n_2 is zero, then definitely, count of bigram is zero due to non-occurrence. Unseen bigrams and unigrams can be avoided using Kneser-Ney and Good Turing smoothing respectively. If smoothing is not applied, and default brackecting is chosen for the sequence, then it can add lot of noise. Therefore, for methods described below, smoothing and normalized pointwise mutual information has been used with lexical association and jaccard index for conceptual association. Below, we have discussed methods depending on various ways of calculating association value.

3.1 Lexical Association using Synonymns and Similar Words

It is not even too easy to find head of two subsequences together in a corpus. To increase the chances of finding the correct parse, we use the synonyms of the head of sub-sequences. Synonyms and similar words can be found using Hindi Wordnet. Synset can give set of synonyms. And similar words can be obtained using is-a specialization or hypernymy tree. Two words are similar to each other, if they share a hypernym. In English Wordnet, this is know as *co-ordinate term*. Example: "kuttA" (dog) and "nevalA" (mongoose) are co-ordinate terms as they both have a common hypernym "carnivore". Association using synset and coordinate terms can be found in various ways which are discussed below.

Sum of Frequency of Synonyms

For this method, we have used association similar to normalized pointwise mutual information. This depends on sum of frequency of all combinations formed using synonyms of both head nouns.

$$AV_{\text{freq\_sum}} = \frac{\ln \frac{sum(n_1,n_2)*N}{sum(n_1)*sum(n_2)}}{-\ln \frac{sum(n_1,n_2)}{N}} \quad (6)$$

$$CV = \sum_{\substack{n \in node \\ n \neq leaf\ node}} AV_{\text{freq\_sum}}(H(lc(n)), H(rc(n))) \quad (7)$$

$sum(n_1, n_2) = \sum_{\substack{s_1 \in synonym(n_1) \\ s_2 \in synonym(n_2)}} freq(s_1, s_2)$,
$sum(n_1) = \sum_{s_1 \in synonym(n_1)} freq(s_1)$,
$sum(n_2) = \sum_{s_2 \in synonym(n_2)} freq(s_2)$

Different authors can write a word with variations. Example, word "gehUZ" (wheat) can also be written as "geMhU". In synset, these variations are available in Hindi Wordnet. Using this approach for association, these variations can be captured. But problem with this method is that number of synonyms for the two head nouns may vary and hence, association value may be misleading.

Sum of Lexical Association of Synonyms

For this method, association in between two heads is calulated using summation of lexical associations between the pairs formed using synonyms of head constituents.

$$AV_{\text{LA\_sum}}(n_1, n_2) = \sum_{\substack{s_1 \in synonym(n_1) \\ s_2 \in synonym(n_2)}} AV_{\text{npmi}}(s_1, s_2) \tag{8}$$

$$CV = \sum_{\substack{n \in node \\ n \neq leaf\ node}} AV_{\text{LA\_sum}}(H(lc(n)), H(rc(n))) \tag{9}$$

This method suffers from the problem of collocation and different number of synonyms. Example for collocation: "vidhAna sabhA" (legislative assembly) is not referred as "kAnUna sabhA", despite the fact that "kAnUna" is the synonym of "vidhAna". Therefore, this method is not a good one.

Maximum of Lexical Association of Synonyms

For this method, association value is the maximum of all the values obtained using the combination of head noun synset.

$$AV_{\max}(n_1, n_2) = \max_{\substack{s_1 \in synonym(n_1) \\ s_2 \in synonym(n_2)}} AV_{\text{npmi}}(s_1, s_2) \tag{10}$$

$$CV = \sum_{\substack{n \in node \\ n \neq leaf\ node}} AV_{\max}(H(lc(n)), H(rc(n))) \tag{11}$$

This method does not have disadvantage of collocation, as we are trying to choose the best possible option from all the association values. Also, it does not even suffer from the problem of different number of synonyms. But disadvantage of this approach is the fact that different variations of writing a word is not taken into account.

3.2 Conceptual Association

Association between two constituents can also be measured using noun classes. We have used Hindi WordNet for finding the noun category and Jaccard Index as the association measure. Instead, of frequency of noun constituents, frequency of noun class is used which is obtained from training data. We have experimented using four types of classes: top node from hypernymy tree, second top node from hypernymy tree, second top node from ontology and third top node from ontology. First node from ontology is not considered as the class, because it tells that a word is noun and all the words in a sequence are noun except genitives. Therefore, it does not give any additional information.

$$CV = \sum_{\substack{n \in node \\ n \neq leaf\ node}} AV_{\text{ji}}(class_{\text{l}}(n), class_{\text{r}}(n)) \tag{12}$$

$$AV_{\text{ji}} = \frac{A}{A + B + C} \tag{13}$$

where,
$class_{\text{l}}(n)$ gives the class of head noun of left child
$class_{\text{r}}(n)$ gives the class of head noun of right child
$A = freq(c_1 c_2)$
$B = freq(c_1 \bar{c}_2)$
$C = freq(\bar{c}_1 c_2)$
and, c_i is the noun class of n_i

Author can write two noun compounds in various manners like dash can be used in between two nouns, two nouns can be combined together etc. Whenever, this feature is found for the two head nouns, then for that pair, one is added to the association value. Adding one, gives extra weightage to that pair and hence shows that these two should be grouped together. This feature is not used with lexical association because such cues were found to be very less frequent in the corpus.

3.3 Backoff Association

Due to less coverage of words in lexical database, conceptual association may not help in predicting the parse. If for any of the parse tree, any of the head noun is not found in lexical database, then normalized pointwise mutual information can be used as the lexical association measure. For lexical association, smoothed frequency is used. Synonyms and similar words based methods are not considered because of the non-significant difference in the result.

4 Bracketing using Hybrid Method

As it is evident from context-free grammar of noun sequences, compound nouns should be bracketed first and then these can be grouped with another compound noun or noun separated by genitives. Since, we use modificational structure for the task of parsing. Therefore, after the step of grouping compound nouns, whole sequence should be grouped using head nouns of these sub-sequences. If the compound noun has more than two nouns, then that sub-sequence is grouped internally using backoff association. Example:

hindU samudAya ke logoM kI bhAvanAyeM
"Hindu" "community" "peoples" "emotions"

In this sequence, hindU samudAya is the compound noun, therefore it should be grouped first as in "(hindU saumUdAya) ke logoM kI bhaAvanAyeM". Now, as the next step "samudAya ke logoM kI bhAvanAyeM" should be parsed.

Head nouns are grouped using agreement. If agreement is not satisfied, then the corresponding parse tree from all possible trees is discarded. In "samudAya ke logoM kI bhAvanAyeM", there are two possible bracketing options: "((samudAya ke logoM) kI bhAvanAyeM)" and "(samudAya ke (logoM kI bhAvanAyeM))". For the first case, "ke" and "logoM" should be in agreement and for second case, "ke" and "bhAvanAyeM" should be in agreement. "ke" and "logoM" are masuline while "bhAvanAyeM" is feminine. And, hence for the second case, "ke" and "bhAvanAyeM" are not in agreement. Therefore, this option should be discared. Now only one option is left: "((samudAya ke logoM) kI bhAvanAyeM)". Therefore, whole parse structure is: "(((hindU samudAya) ke logoM) kI bhAvanAyeM)". There can be the cases, when more than one tree is left in option after the process of removal of trees. Then, for those trees, we find cohesion value using backoff association. If none of the parse tree is possible to be formed, then it has the ability to tell about grammatical error.

Syntactic rules are not used directly as done by Batra et al. (2014) and Sharma (2011), because they are valid for the presence of two genitives. We know that noun sequence can be of many possible length with different number of genitives. Therefore, it becomes difficult to increase the number of rules and hence we have used this procedure.

5 Experiments, Results and Observations

For experiments, 2365 noun sequences were extracted from Hindi Treebank. We have taken sequences with maximum five components because of less occurrence of larger sized sequences. Distribution of noun sequence on the basis of number of noun constituents is shown in Table 1. Since, conceptual association based approaches require training data, the data of noun sequences is divided into 2:1 ratio for training and testing respectively.

Noun Count	Distribution
3	78.30%
4	17.37%
5	3.55%
6	0.50%
7	0.25%

Table 1: Percentage distribution of noun sequences according to number of noun constituents

Batra et al. (2014) have shown that there exists more cases of left bracketing than right bracketing. Therefore, for all of the approaches, if two or more bracketing option has same cohesion value, then one with left bracketing is chosen out of the conflicts.

Further, a corpus of size 18200k, obtained using web crawling was used for finding frequency of noun constituents. Frquency is found for root form of the head nouns. And since, frequency for root form is used, therefore, root form of corpus is also used for finding the frequency. Also genitive construction as the paraphrase is used for

increasing the bigram counts. For variations in lexical association, first we have experimented using different association measures with smoothed frequency. It has been observed that normalized pointwise mutual information performs the best. Also, three variations for lexical association depending on synonyms and similar words have been used. It has been observed that approaches using similar words is not performing as good as the one with only synonyms. A lot of noise is added for similar words due to the problem of collocation. For the same reason, summation of lexical association method is performing the worst. Even method based on summation of frequency is not performing too good. It is adding more noise due to difference in number of synonyms. The fact that it can be useful for capturing variation of spelling is overshadowed. Results for all these variations are shown in Table 2.

Type	Accuracy
jaccard index	58.49%
normalised pmi	59.00%
chi-square	58.23%
synset + frequency sum	58.49%
synset + association sum	45.08%
synset + association max	61.43%
similar + frequency sum	48.91%
similar + association sum	46.99%
similar + association max	49.04%

Table 2: Accuracy for methods using Lexical Association

For conceptual association, experiments have variation in terms of noun class. It has been observed that noun class obtained from hypernymy tree is performing better than the one obtained using onotology. Number of noun classes from hypernymy tree are greater than ontology. Therefore, it is able to capture variations of noun sequences as much as possible. Also, second top node of hypernymy is not performing better than the top node and third top node of ontology is performing worse than the second one. Inspite of the fact that number of noun classes are large, the accuracy is not good. Many times, no class is assigned because of small depth of hypernymy and ontology tree. For the association measure, normalised pointwise mutual information and jaccard index is used for all these methods. It has been seen that top hypernymy node is performing best with jaccard index. Then, the feature of dash is taken into consideration for this approach. Results for all these variations are shown in Table 3.

Type	Accuracy
ji + 2^{nd} top ontological node	50.70%
ji + 3^{rd} top ontological node	48.02%
ji + top hypernymy node	59.64%
ji + 2^{nd} top hypernymy node	54.91%
npmi + 2^{nd} top ontological node	57.72%
npmi + 3^{rd} top ontological node	55.93%
npmi + top hypernymy node	57.47%
npmi + 2^{nd} top hypernymy node	57.47%
ji + top hypernymy + dash feature	60.66%

Table 3: Accuracy for methods using Conceptual Association

Hindi Treebank has the collection of news. Many english words can be found written in Hindi. It has been observed that these words have better chances to be found in corpus than lexical database. Therefore, for such cases, lexical association is working better than conceptual association.

For further experiments, cohesion value is obtained using backoff association. Then experiment is done with grouping compound nouns and bracketing head nouns using backoff association without considering agreement. As the part of the last experiment, agreement is also used. Results are shown in Table 4.

Type	Accuracy
backoff association (BA)	61.55%
BA + CN grouping	81.48%
BA + CN grouping + agreement	86.33%

Table 4: Accuracy for methods using Backoff Association

For baseline, left brcketing is applied. In Table 5, results for 3, 4 and 5 components are shown for left bracketing and the hybrid approach to show that these experiments are performing bettter than the baseline. Left bracketing is the parse tree of n-1 height and ‘n’ nodes. In this tree, every non-leaf node has the right child which is a leaf node. Example: ((((n1 n2) n3) n4), ((((n1 n2)n3)n4)n5) etc. As the number of constituents increases, number of possible bracketing options increases. Hence, the task becomes more and more difficult

and this is also evident from the results shown in Table 5.

Noun Count	Left Bracketing	Hybrid
3	76.16%	92.16%
4	45.23%	68.25%
5	25.00%	43.75%

Table 5: Accuracy for left bracketing and hybrid approach

6 Conclusion

Each type of association has some advantages. Lexical association works good, if corpus size is big and unbiased. Similar words should be avoided as it adds lot of noise. Conceptual association has advantage of learning data. Increasing the learning data can help in improving the power of conceptual association. And when both associations are combined, then problem of less coverage of words in lexical database can be overcome for conceptual association. When problems of both association are removed, accuracy for methods using only semantic knowledge can be improved as shown for English compound nouns. Knowing about syntax of noun sequence and agreement feature also helps. Therfore, hybrid approach is generally good for the cases when small size of resources are available. For future work, spelling normalizer can be applied on noun sequences for improving methods using lexical association.

References

Amba Kulkarni and Anil Kumar. 2011. Statistical constituency parser for sanskrit compounds. *Proceedings of ICON*.

Arpita Batra, Soma Paul, and Amba Kulkarni. 2014. Constituency parsing of complex noun sequences in hindi. In *Computational Linguistics and Intelligent Text Processing*, pages 285–296. Springer.

Dan Jurafsky and James H Martin. 2000. *Speech & language processing*. Pearson Education India.

Donald Hindle and Mats Rooth. 1993. Structural ambiguity and lexical relations. *Computational linguistics*, 19(1):103–120.

James Pustejovsky, Peter Anick, and Sabine Bergler. 1993. Lexical semantic techniques for corpus analysis. *Comput. Linguist.*, 19(2):331–358, June.

Kobayasi Yosiyuki, Tokunaga Takenobu, and Tanaka Hozumi. 1994. Analysis of japanese compound nouns using collocational information. In *Proceedings of the 15th conference on Computational linguistics-Volume 2*, pages 865–869. Association for Computational Linguistics.

Mark Lauer. 1994. Conceptual association for compound noun analysis. In *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*, pages 337–339. Association for Computational Linguistics.

Mark Lauer. 1995. Corpus statistics meet the noun compound: some empirical results. In *Proceedings of the 33rd annual meeting on Association for Computational Linguistics*, pages 47–54. Association for Computational Linguistics.

Mark Lauer and Mark Dras. 1994. A probabilistic model of compound nouns. *arXiv preprint cmp-lg/9409003*.

Mark Liberman and Richard Sproat. 1992. The stress and structure of modified noun phrases in english. *Lexical matters*, pages 131–181.

Mirella Lapata and Frank Keller. 2005. Web-based models for natural language processing. *ACM Transactions on Speech and Language Processing (TSLP)*, 2(1):3.

Mitchell P Marcus. 1980. *Theory of syntactic recognition for natural languages*. MIT press.

Philip Resnik. 1992. Wordnet and distributional analysis: A class-based approach to lexical discovery. In *AAAI workshop on statistically-based natural language processing techniques*, pages 56–64.

Philip Resnik and Marti Hearst. 1993. Structural ambiguity and conceptual relations. In *Proceedings of the Workshop on Very Large Corpora: Academic and Industrial Perspectives*, pages 58–64. Citeseer.

Preslav Nakov and Marti Hearst. 2005. Search engine statistics beyond the n-gram: Application to noun compound bracketing. In *Proceedings of the Ninth Conference on Computational Natural Language Learning*, pages 17–24. Association for Computational Linguistics.

Ramakanth Kavuluru and Daniel Harris. 2012. A knowledge-based approach to syntactic disambiguation of biomedical noun compounds. In *COLING (Posters)*, pages 559–568. Citeseer.

Roxana Girju, Dan Moldovan, Marta Tatu, and Daniel Antohe. 2005. On the semantics of noun compounds. *Computer speech & language*, 19(4):479–496.

Sapna Sharma. 2011. *Disambiguating the parsing of hindi recursive genitive constructions*. Ph.D. thesis, International Institute of Information Technology Hyderabad–500032 India.

Yiming Yang and Jan O Pedersen. 1997. A comparative study on feature selection in text categorization. In *ICML*, volume 97, pages 412–420.