Automatic Annotation of Genitives in Hindi Treebank

Nitesh Surtani, Soma Paul

Language Technologies Research Centre
IIIT Hyderabad
Hyderabad, Andhra Pradesh-500032
nitesh.surtaniug08@students.iiit.ac.in, soma@iiit.ac.in

ABSTRACT

Noun with genitive marker in Indo-Aryan language can variously be a child of a noun, a verb or a complex predicate, thus making it an important parsing issue. In this paper, we examine genitive data of Hindi and aim to automatically determine the attachment and relational label of the same in a dependency framework. We implement two approaches: a rule based approach and a statistical approach. The rule based approach produces promising results but fails to handle certain constructions because of its greedy selection. The statistical approach overcomes this by using a single candidate approach that considers all the possible candidates for the head and chooses the most probable candidate among them. Both approaches are applied on controlled and open environment data. A Controlled environment refers to the situation when the relational labels are attested to the input data except for the genitive data; while open environment refers to cases in which the input is only POS tagged and chunked. The rule based and statistical systems produce a high accuracy of 95% and 97% respectively for attachment and perform considerably well for labeling in controlled environment but poorly in open environment.

KEYWORDS: TREEBANK, GENITIVE, DEPENDENCY RELATION, ATTACHMENT, LABELING, SINGLE CANDIDATE

Proceedings of the Workshop on Machine Translation and Parsing in Indian Languages (MTPIL-2012), pages 1–14,
COLING 2012, Mumbai, December 2012.

1 Introduction

Nouns with genitive case marker occur in various syntactic contexts in Indo-Aryan languages. The default genitive case marker specifies a relation between two nouns: head and modifier as in raama kaa ghara *(Ram's house), pital*a kaa bartana (utensil of copper) etc. where raama and pitala (copper) are modifiers that modify the head ghara (house) and bartana (utensil) respectively[1]. Genitive nouns are also found to occur in many other contexts. The most significant one is the relation that occurs between the genitive noun and verb as illustrated in (1).

1. raama ke do bete hain
   Ram gen two sons be-3pl pr
   'Ram has got two sons.'

The genitive in (1) is distinct from the one illustrated in (2) which is a regular noun-noun genitive.

2. raama ke do bete skula jaa rahe hain
   Ram gen two sons school go be-3pl pr
   'Two sons of Ram are going to school.'

A genitive can occur with a complex predicate. Complex predicate in Hindi is a construction that is composed of a noun or an adjective and a light verb. For example, pratikshaa karnaa in (3) is a complex predicate because the construction is a multiword expression that denotes a single event. As noted in (3), the two arguments of the complex predicate are raama and sitaa, the latter one being cased marked for genitive. Here the genitive marked noun is the theme argument of the verb.

3. raama sitaa kii pratikshaa kara rahaa thaa
   Ram Sita gen wait do be-3sg pst
   'Ram was waiting for Sita.'

The argument of a verb regularly takes a genitive in the context of a verbal noun[2] form of a verb. In (4), raama is an argument of the verb jaa 'go'.

4. raama kaa jaanaa sambhava nahii hai

[1] The genitive case marker is kaa, which has allomorphic variations as kii and ke. The allomorphic variation is governed by the grammatical features of the head noun as illustrated below:

Genitive allomorph	Head grammatical feature	Example
kaa	Masculine, Singular, Direct Case	raama kaa ghara 'Ram's house'
ke	Masculine, Singular, Oblique Case	samwaadaataa ke sawaala kaa javaaba diyaa 'Answered the question of Press'
	Masculine, Plural, Any Case	congress ke vaade 'Promises of congress'
kii	Feminine, Any	brahaspatiwaara kii raata 'Thursday's night'

[2] In Hindi, verbal noun form of a verb is derived by adding a suffix –ne to the verb as in: jaa 'go' → jaanaa 'going', likh 'write' → likhnaa 'writing' etc.

Ram gen go-VN possible neg be-3sg pr
'It is not possible for Ram to go.'

With a verbal noun form, the argument is marked with genitive case. The same holds even when some participants intervenes the two as illustrated in (5). The argument raama is separated from jaanaa with two other participants, sitaa 'Sita' and ghara 'home'.

5. raama kaa sitaa ke saatha ghara jaanaa sambhava nahii hai
Ram gen Sita gen with home go-VN possible neg be-3sg pr
'It is not possible for Ram to go home with Sita.'

Apart from the above cases, one significant occurrence of genitive is when the head is elided as illustrated in (6).

6. yaha khaanaa kala kaa hai
This food yesterday gen be-3sg pr
'This food is yesterday's (food).'

We have examined various distributions of genitive data in Hindi. Table 1 attempt to tabulate all the types of genitive that we have discussed in this section:

CASE	CONSTRUCTION TYPE	EXAMPLE
Case 1	Noun gen – Noun	raama kaa ghara 'Ram's house'
Case 2	Noun gen – Verb	raama kaa eka betaa hai 'Ram has one son'
Case 3	Noun gen – Complex predicate	raama sitaa kii pratikshaa kara rahaa thaa 'Ram was waiting for Sita'
Case 4	Noun gen – Verbal Noun	raama kaa jaanaa 'Ram's leaving'
Case 5	Noun gen – Head elided	yaha khaanaa kala kaa hai 'This (food) is yesterday's food'

TABLE 1: Different type of genitive data in Hindi

Thus, even though a genitive noun by default modifies a noun, it also occurs in other contexts including relation with verbs, with complex predicates and so on. This amounts to a great parsing issue of how to determine the correct relation for a genitive modifier in a sentence. In the context of dependency parsing, the task is twofold: 1. Determining the attachment of the genitive modifier with its legitimate head; 2. Predicting the correct relation for the attachment. The relation labels are adopted from those used in Hindi syntactic Treebank (Bharati et. al., 2009a). We implement two systems: (A) A Rule-Based system: which implements the cues as rules for predicting the correct attachment between genitive modifier and its head and (B) A Statistical system: which uses a single candidate approach; which considers all the possible candidates for the head and chooses the most probable candidate among them as the head. The rule based system has a drawback of making a greedy choice. The single candidate approach overcomes this drawback and shows to outperform the rule based system by achieving an accuracy of 97%, in contrast to the accuracy of 95% achieved by the rule based system. The results are quite encouraging and a lot of human labor and time can be saved if such data is automatically labeled for correct relation most of the time.

The paper is divided into the following sections. Section 2 talks about the related works. Section 3 presents a brief overview of Hindi Treebank, an annotated corpus resource that we have used for the present work and presents a study on the distribution of genitives in Hindi Treebank. Section 4 talks about the automatic annotation approaches and describes the experimental setup. Section 5 and 6 discuss the implementation of the rule based and the statistical system for automatic labeling of the genitive data along with the data preparation, parameters and results of the corresponding systems. Section 6 concludes the paper.

2 Related Works

A syntactically annotated treebank is a highly useful language resource for any NLP task and the correctness of the annotated data is very important. Generally, building a treebank requires an enormous effort by the annotator. Some research has been done in the direction of semi-automating the Treebank annotation (Lim et al., 2004). This, on one hand reduces the human effort by decreasing the number of intervention required by the annotator, and helps in maintaining consistent annotation in building a Treebank on the other.

Gupta et al. (2008) attempts a rule based approach for the automatic annotation of the Hindi Treebank and labels a set of coarse grained kaaraka and non-kaaraka relations. It identifies genitives (r6) with an f-score of 82.1% for correct attachment and labeling. Hybrid approaches (Bharati et.al, 2009b) and statistical approaches have also been attempted for automatic parsing using Hindi Treebank. Malt Parser (version 0.4) (Nivre et al., 2007), and MST Parser (version 0.4b) (McDonald et. al., 2005) have been tuned for Hindi by Bharati et al. (2008). Kosaraju et.al (2010) reports an accuracy of 87.03% for the correct attachment and labeling of the genitive data using the malt parser. Table 2 shows the results obtained using Malt parser:

Label	Accuracy
r6	87.03
k1	81.92
k2	72.80
pof	84.10

TABLE 2: Results of the baseline system

We use the above result as the baseline result for our experiments.

3 Genitives in Hindi-Urdu Treebank

This section presents a brief description of the Hindi-Urdu Treebank followed by the distribution of genitive data as attested in the Treebank.

3.1 Brief description of Hindi-Urdu Treebank

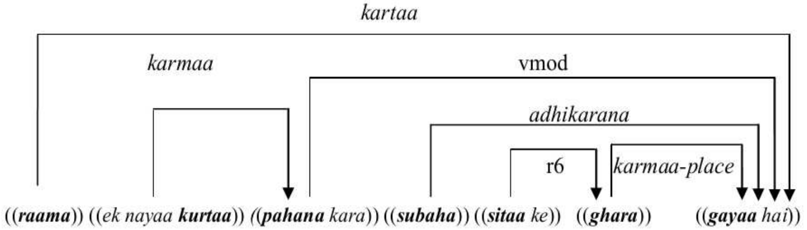
The Hindi-Urdu dependency Treebank is being developed following the analysis of the Paninian grammatical model (Bharati et al., 2009a). As observed in Bhatt et al. (2009), “the model offers a syntactico-semantic level of linguistic knowledge with an especially transparent relationship between the syntax and the semantics.” At present there are 10799 sentences (of around 250 thousand words) that are annotated with dependency relations. The dependency relations are of two types: kaaraka and non-kaaraka. kaaraka relations indicate the roles that various participants play with respect to a verb. Every kaaraka relation has a well-defined semantics as described in the Paninian Grammar. There are six kaaraka relations: kartaa (k1), karmaa (k2), karana (k3), sampradaana (k4), apaadaana (k5) and adhikarana (k7). Even though attempts are being made to relate these relations to richer semantic roles of VerbNet and FrameNet via Propbank (Bhatt 2009), kaaraka relations capture one very significant semantic-pragmatic information which is known as vivakshaa that can be translated as 'speaker's choice'. For example, the subject of the following sentence is marked as kartaa although it is a 'theme' in terms of its semantic role:

kartaa
7. darwaazaa khulaa
door open-3p pt
'Door opened.'

Semantics of these relations are given in details in Bhatt et. al (2009). In this approach, sentences are treated as a series of chunks with every chunk having a head and one or more optional modifier of the head. For example, the chunks of the following sentence are shown below. The head of each chunk is highlighted

8. ((**raama**)) ((ek nayaa **kurtaa**)) ((**pahana** kara))((**subaha**))((**sitaa** ke)) ((**ghara**)) ((**gayaa** hai))
Ram one new **shirt** **wear**-3sg pr **morning** **Sita** gen house go-perf be-3p pr
'Ram went to Sita's house in the morning wearing a new shirt'

The main verb is taken to be the head of the sentence and all other chunks are connected to the head through appropriate relations. Genitive modifiers (as sitaa ke 'of Sita' in (8)) are generally attached to nouns and the relation is labeled as r6 (see section 3.2 for more details). A noun can occur in kaaraka relation with a verb if it has a direct syntactic relation with its head verb. For example, the relations will be the following for the above sentence:


kartaa
karmaa
vmod
adhikarana
r6
karmaa-place
((raama)) ((ek nayaa kurtaa)) ((pahana kara)) ((subaha)) ((sitaa ke)) ((ghara)) ((gayaa hai))

Relations other than 'kaaraka' such as purpose, reason, and possession are also captured using the relational concepts. For example, in the above sentence, the two verbs gayaa hai which is finite and pahana kara which is non-finite are related with a 'vmod'- relation and captures the information that the non-finite verb is dependent on the finite one.

3.2 Distribution of Genitives in Hindi Treebank

Genitive data is quite frequent in the Hindi Treebank. There are a total of 11505 cases of genitives in a corpus of 10799 sentences (of around 250 thousand words). We note that, as expected, Case 1 (noun genitive-noun) occurs most number of times (9123 out of 11505). As discussed in Surtani et. al. (2012), the relation is tagged with a label called r6 that represents the notion of sashthii sambanadha of the Paninian grammar. The symbol 'r' indicates that it is not a kaaraka relation and the numeral 6 represents sashthii (sixth) relation. The label r6v (Case 2 in Table 1) indicates that the genitive modifier is related to verb and not with any noun as generally is the case with genitives. This label is semantically not very informative, which is the case even with the r6 relation. On the other hand, the labels for Case 3, namely r6-k1 and r6-k2, represent both syntactic and syntactico-semantic level of information. The labels k1 and k2 convey that the noun is kartaa and karmaa respectively and the r6 part indicates that these kaaraka relations are physically represented by a genitive case marker. When the head noun is derived from a verb, the POS tag for such word is given VGNN. The tag implies that the word is a noun derived from a verb. Since, the verbal noun forms retain the verbal property[3] the genitive modifiers of these nouns are tagged with kaaraka relation. Following examples indicate that the genitive can be kartaa (see (9)) or karmaa (see (10))

9. party kaa kahanaa hai…
   party gen say-VN be-3pr sg
   'It is what Party has to say that…'

10. aatankiyon ke maare jaane kii sankhyaa...
    terrorists gen being killed gen number
    'The number of terrorists being killed …'

From the treebank, we come to know that the genitive karmaa (k2) of a verbal noun is much rarer than the genitive kartaa (k1) as recorded in the following table in Case 4. Table 3 presents distribution of different genitive types in Treebank. We have listed those relations which have at least 5 occurrences for genitive noun in the Treebank.

CASE	Construction Type	Relation Label	No. of occurrence	%
Case 1	Noun gen – Noun	r6	9123	79.65
Case 2	Noun gen – Verb	r6v	16	0.14
Case 3	Noun gen – Complex predicate	r6_k1	337	2.94
		r6_k2	1595	13.93
Case 4	Noun gen – Verbal Noun	k1	370	3.23
		k2	13	0.11

TABLE 3: Distribution of genitive data in Hindi Treebank

In the default order of genitive construction in Hindi, the genitive modifier precedes its head. But, Hindi being a free word-order language, we come across cases in the Treebank, where the genitive modifier occurs after the head, which we term here as 'Marked order'. We study the occurrence of 'Marked order' data in Treebank and notice that such data is very rare in the Treebank. There are 37 instances of 'Marked order' data out of total of 11505 cases (approx 0.32 % of times) of genitives in Treebank.

Since the occurrence of marked order data is very less, we neglect it and consider only the data in default order for our experiments. A genitive noun is contiguous with its head if the position of

[3] A verbal noun licenses all participants of the base verb and the vibhaktii or case markings on the participants are also retained except for kartaa and karmaa which is generally expressed by genitive case marker. Thus the verbal noun form of the verb socha 'think' is sochnaa 'thinking' licenses the participants as illustrated below: tumharaa isa vishaya para aisaa sochnaa galata nahii thaa. The karta is marked with genitive case as in tumharaa, but the adhikarana kaaraka (or subject matter) is expressed with 7th case ending as would have been the case, when the verb form occurs as in: tuma ne isa vishaya para jo sochaa wo galata nahii thaa.

the head is next to the genitive noun. Table 4 presents the contiguity statistics of the genitive data. The Non-contiguous case with an intervening candidate specifies that a noun, a verbal noun or a verb (i.e. a legitimate head candidate) falls between the head and the genitive modifier. The case is Non-contiguous with no intervening candidate if the genitive modifier is not contiguous with its head and no head candidate occurs in between the genitive noun and the head.

CASE	**Construction Type**	**Relation Label**	**No. of occurrence**	**Contiguous**	**Non-Contiguous (With intervening candidate)**	**Non-Contiguous (Without intervening candidate)**
Case 1	Noun gen – Noun	r6	9123	8642 (94.73)	453 (4.96%)	28 (0.33)
Case 2	Noun gen – Verb	r6v	16	10 (66.66%)	3 (14.28%)	3 (19.04%)
Case 3	Noun gen – Complex predicate	r6_k1	337	310 (91.98%)	21 (6.2%)	6 (1.8%)
		r6_k2	1595	1429 (89.58%)	144 (9.06%)	22 (1.36%)
Case 4	Noun gen – Verbal Noun	k1	370	289 (78.34%)	48 (12.96%)	33 (8.7%)
		k2	13	7(48.15%)	4 (44.44%)	2 (7.4%)
Total			11454	10687	673	94

TABLE 4: Contiguity statistics

The occurrence of contiguous data in the Hindi Treebank is quite high. This motivates us to build a Rule based system for the automatic annotation of the genitive data. The next section discusses the systems for automatic labeling of the genitives.

4 Automatic labeling of Genitive data

Manual development of Treebank is a time consuming and labor intensive task. Attempts have been made by Gupta et.al (2008), Lim et.al (2004) to automate some part of the task so that data development becomes fast. Our attempt is to predict the correct attachment for a genitive noun and mark the relation label between the genitive noun and its head. A survey of genitive data in Hindi Treebank motivates us towards developing a rule based system for automatic annotation of the Hindi genitive data. The system performs quite well because of the contiguous nature of the genitive data in Hindi. Although it is handling most of the cases in the data, it is unable to handle certain constructions especially the ones that are non-contiguous. The main reason for this can be attributed to the greedy selection made by the rule based system as it chooses the first liable candidate, the one that satisfies the rules, as the head of the genitive marked chunk. Thus, it fails to consider all the competing head candidates and choose the best candidate from them. To overcome this issue, we use a single candidate approach which chooses the most probable head from all the competing candidates. We have implemented both the systems in two environments:

(i) **Controlled Environment**: In this scenario, all the other dependency relations, except for the genitive are marked in the sentence. The system uses this information to predict the correct attachment and the syntactic-semantic label between the genitive child and its head.

(ii) **Open Environment**: In this situation, the input data is only POS tagged and chunked. The system has no information about the relational labels of other chunks.

The information about the kaaraka label and complex predicate is essential for predicting the correct labels of Case 3 (Noun gen-Complex Predicate). But identifying these labels is a parsing issue in itself. Thus, the accuracy of labeling the head-modifier drops down significantly in the open environment as this information is in this environment. Similarly, the systems are unable to predict the correct syntactico-semantic labels for Case 4 (Noun gen-Verbal noun) in both the environments. Next sections discuss the rule based and the statistical approaches for automatic parsing of the genitive construction.

5 Rule Based System

A survey of the genitive data in Hindi Treebank provided us with syntactic cues for determining the legitimate head of the genitive modifier and the corresponding relation between the two. This motivates us to developing a rule based system by implementing these cues as rules for automatic annotation of the Hindi genitive data. We make the following observations from the data that we have studied in section 3.2:

a. A genitive marked noun can only take a noun, a verbal noun or a verb as its head. Therefore, the remaining POS categories are not the probable candidates for head of a genitive modifier.
b. The case of head nouns modified by a genitive noun is the most frequent and regular one in the treebank.
c. The head of the genitive modifier is mostly contiguous to the modifier. As illustrated in Table 4, the head occurs next to the genitive noun 94.73% of the time.
d. The genitive case marker gets its grammatical features from its head. Therefore, there is a grammatical agreement in the features of the head and the genitive case marker.
e. A genitive noun cannot have a pronoun (as the head of the noun chunk) as its head.
f. Once a noun identified as part of complex predicate, a genitive noun modifier of that noun will regularly be in r6_k*. However, it is difficult to determine the correct kaaraka relation from the surface cues alone.
g. The number of occurrences of genitive modifiers with a direct verb (i.e. r6v) is few compared to other kinds of genitive construction.
h. Genitives that modify verbal noun indicate different kaaraka relations (see table 3)

5.1 Data

The rule based system is tested on the default order test data of 11454 genitive instances. Since ‘Marked order’ data is very less in the Treebank, such data is ignored in the present experiment. We have also not included data for genitive modifiers that modify non-finite verb because of non-representativeness of such data in the Treebank.

5.2 Implementation

A set of rules have been crafted and implemented for identifying the right attachment and syntactico-semantic label for each attachment in the test data. The rules basically verify whether an NP chunk with a genitive case marker within is followed by a Noun phrase, a Verb phrase or a Verbal noun phrase.

1) The system assigns the relations r6, r6v and k* respectively if the Noun phrase with genitive case marker is followed by a Noun phrase, Verb phrase or a Verbal Noun phrase. In case the genitive modifies a complex predicate (i.e. the head of the modifier occurs with ‘pof’ relation with a light verb), the genitive noun is labeled with r6_k*.

2) The agreement of the following morphological features of the child and the head are matched. All these features must agree for the candidate to be the liable head of the child:

a. Gender: Gender can be masculine (m) or feminine (f). It takes value 'any' in case it can be of any of the forms.

b. Number: Number can be singular (sg) or plural (pl).

c. Person: It can be 1st Person (1), 2nd person (2) or 3rd person (3).

d. Case: Case can either be direct (d) or Oblique (o). This feature is handled differently for pronoun[4].

3) Head as Pronoun: A genitive marked noun phrase cannot take pronoun as the head.

The rule based system implements these rules to predict the attachment and the label between the head and genitive noun. The system matches the rules for the genitive modifier and the candidate chunk and assigns the candidate chunk as the head of the genitive modifier only if all the rules are matched. The corresponding relational label is then assigned to the head-child pair.

5.3 Result and Observation

The experiments were performed in both the controlled and the open environments. The results are presented below.

CASE	**Relation Label**	**Number of occurrence**	**Attachment**		**Labeling**			
					Controlled Env.		**Open Env.**	
			Frequency	Accuracy	Freq	Acc	Freq	Acc
Case 1	r6	9123	8771	96.14	8771	96.14	8771	96.14
Case 2	r6V	16	13	81.25	13	81.25	13	81.25
Case 3	r6_k1	337	316	93.88	316	93.88	0	0
	r6_k2	1595	1464	91.8	1464	91.8	0	0
Case 4	k1	370	323	87.82	0	0	0	0
	k2	13	8	61.54	0	0	0	0
Total		11454	10895	95.12%	10564	92.23%	8784	76.7%

TABLE 5: Result of the rule based system

The rule based system predicts the head of the genitive modifier with an accuracy of 95.12% in both the controlled and the open environment. As already discussed, the correct syntactico-semantic label in the open environment is predicted with low accuracy. As shown in Table 5, the system is unable to predict the label in Case 3 and Case 4 in open environment and Case 4 in controlled environment, since prediction of kaaraka relation becomes a parsing issue in itself. Though, the kaaraka relations in a sentence can also be predicted with a considerable accuracies, as already shown in Table 2, we do not consider them for our calculations. The accuracy of the system for the labeled attachment drops down to 92.23% in controlled environment and with 76.7% in open environment.

[4] In case of nouns, the child noun and its genitive marker occur as different tokens. The genitive marker obtains its grammatical case information from the head of the genitive marked noun. But in case of pronouns, the pronoun root form is inflected with the genitive case marker to form a single token. Therefore, it always occurs in oblique (o) form and thus, has not been considered for grammatical agreement.

The result is encouraging because, our Treebank has highest number of representation of Case 1 data. If such data can automatically be labeled for correct relation for most of the time, a lot of human labor and time can be saved. Table 5 indicates that the performance for genitive modifier – noun construction is exceptionally good, achieving an accuracy of 96.14%; while for other kind of construction, we achieve a mediocre score because of the high percentage of the non-contiguous occurrences between the genitive noun and its head. The algorithm used in the rule based system is a greedy one in the sense that it will pick up the first context that all the rules satisfy without verifying other contexts. For example, given the following sentence, raama kaa ghara jaanaa 'Ram's going home', the system will connect raama 'Ram' with ghara 'home' and assign an r6 label without considering the possibility of raama's being connected to jaanaa 'go' which would be the right attachment in this case. Thus, a rule based system fails to consider all the candidates for the head of the modifier. Therefore, a model that considers all the candidate heads and selects the most probable head from all the competing candidates should work better for handling this issue. A single candidate approach is tried out for this which is discussed in the next subsection.

Label	**r6**	**r6v**	**r6_k***	**k***
r6	9102	0	12	9
r6V	3	13	0	0
r6_k1	10	5	316	6
r6_k2	93	24	1464	14
k1	44	1	2	323
k2	5	0	0	8

TABLE 6: Confusion Matrix of the rule based system

Table 6 represents the number of times each case is labeled by the rule based system. The columns specify the label given by the system. Although, the Case 1 is attached correctly only 8771 times (as shown in table 5), it is given the label r6 9102 times (as shown in Table 6). This is because the attachment of the child is with the wrong NP chunk.

6 Statistical System

As discussed in the previous subsection, the rule based system fails to perform well on the non-contiguous data because of its greedy selection. Therefore, we need a model that considers all the possible candidates for the head of the genitive marked NP and then choose the most probable head among all the candidates. We use a single candidate approach (Yang et.al (2005), Niyu et.al (1998)) using an SVM classifier for predicting the most probable head for the attachment.

6.1 Single Candidate Approach:

The single-candidate approach is a machine learning method, which chooses the most probable candidate from a set of all possible candidates. So, given the child (i.e. the genitive marked NP) and n candidates for heads (C_1, C_2,…,C_n), the model obtains the probability that candidate C_k is the head of the child in context of all other candidates. The single-candidate model assumes that the probability that C_k is the head is only dependent on the child and the candidate C_k, and is independent of all the other candidates.

$$p(head(Ck)|child, C1, C2, .., Cn) = p(head(Ck)|child, Ck)$$

The single candidate approach is used with an SVM classifier to predict the correct head (C_k).

6.2 SVM Classifier:

Support vector machines, (Vapnik, 1995), are computational models used for the classification task in a supervised learning framework. They are popular because of their good generalization ability, since they choose the optimal hyperplane i.e. the one with the maximum margin and reduce the structural error rather than empirical error. We have used the LIBSVM library (Chang and Lin, 2011) for our task.

6.3 Data Preparation:

In the single-candidate model, an instance has the form {child, head}, where child is the genitive modifier and head is a legitimate head candidate. For training, instances are created for each child occurring in an annotated text. Specifically, given a child and its head candidates, a set of negative instances (labeled "0") is formed by pairing child and each of the candidates that are not the head of the genitive modifier. In addition, a single positive instance (labeled "1") is formed by pairing child and the correct head. Table 7 illustrates the generation of the training instances. In this way, a total number of 38556 instances are created with 11454 positive and 27102 negative instances.

Example: [**dhonii kaa**]/NP [tossa]/NP [jeeta kara]/VGNF [pehle ballebajii]/NP
Dhoni-gen toss win 3pr.non-fin first batting
[karnaa]/VGNN [sahii siddha huaa]/VGF
do-VN right proved be-perf
'Dhoni's winning the toss and electing to bat first proved to be right.'

Instance	Label
{ dhonii kaa, tossa }	0
{ dhonii kaa, jeeta kara }	0
{ dhonii kaa, pehle ballebajii }	0
{ dhonii kaa, karnaa }	1
{ dhonii kaa, sahii siddha huaa }	0

TABLE 7: Example of single-candidate training instances

6.3 Feature Selection

Following features have been used in our experiments for training and testing. Since the experiments are carried out in both the controlled and the open environments, therefore the feature vectors formed in these two experiments are different in terms of the information available. The differences in the features used in these environments are also discussed below.

1) Distance: Distance is defined as the number of candidates between the child and the head chunk. It takes an integer value.
2) Grammatical Features: Grammatical feature includes gender, number, person and case as already discussed in the rule based system. It takes value 1 when all the grammatical features for head and child match. Else, it gets a value -1.
3) Pronoun: Whether the head candidate is a pronoun or not. This feature also takes an integer value.
4) Chunk Type: This feature specifies the type of chunk and takes values 1, 2, 3 and 4 for noun phrase (NP chunk), complex predicate (NP-pof chunk), verbal noun phrase (VGNN) and verb phrase (VGF) respectively in case of controlled environment and 1, 2 and 3 for

noun phrase (NP chunk), verbal noun phrase (VGNN) and verb phrase (VGF) respectively in open environment since the complex predicate information is not available in the open environment.

A feature vector comprising of these 4 features is formed for each instance of the training and the testing data. The relative significance of each feature for the learning model is presented Table 8. The feature for which the performance of the learning model is affected the most, when it is removed from the feature vector, is a more important feature for the model. A feature is pruned at each iteration and the corresponding performance of the model is recorded. We find the relative significance of each feature by pruning one feature each time. The removal of the distance feature from the model reduces its accuracy to 63.67% from the baseline accuracy of 96.86% and hence is most important feature for the model.

Features	**Distance**	**Agreement**	**Pronoun**	**Chunk Type**
Accuracy	63.67%	92.13%	96.86%	96.70

TABLE 8: Relative Significance of features in statistical model

6.4 Training and Testing

While training, the feature vector for each instance is computed and is given input to the SVM classifier along with its label. The classifier learns a model (optimal hyperplane) from the training data. Both the training and the testing data are scaled before the experiment. Grid search is used to find the optimal parameters for learning the model. The total number of instances generated by the single candidate approach is 38556, with 11454 positive instances and 27102 negative instances. We use K-fold cross validation keeping k=5, i.e. dividing the data into 5 folds, where 1 fold is held out for testing while the rest are used for training the model in each iteration. While testing, the model predicts the label of instance, 1 if model predicts that the candidate is the head of the genitive marked NP chunk; -1 otherwise. Since k-fold cross validation technique is used, the model is tested on the complete dataset and we obtain the label for each instance.

6.5 Result and Observation

The results of statistical system for prediction of attachment and the syntactic-semantic label for both the environments are presented below in Table 9. The accuracy of the model in controlled environment is 96.86% as compared to 95.12% in rule based system.

CASE	**Relation Label**	**Number of occurrence**	**Attachment**		**Labeling**	
			Controlled Env.	**Open Env.**	**Controlled Env.**	**Open Env.**
Case 1	r6	9123	97.57	97.70	97.57	97.70
Case 2	r6V	16	87.5	87.5	87.5	87.5
Case 3	r6_k1	337	95.25	93.76	95.25	0
	r6_k2	1595	94.17	93.23	94.17	0
Case 4	k1	370	93.24	92.70	0	0
	k2	13	84.61	76.92	0	0
Total		11454	96.86%	96.76%	93.75%	77.94%

TABLE 9: Result of the statistical system

The accuracy of attachment of the model in the controlled environment is 96.86% as compared to 95.12% in rule based system. The accuracy is reduced by 0.10% in the open environment as the information about the complex predicate is not available. The accuracy of predicting the attachment for Case 3 goes down from 95.25 to 93.76 and 94.17 to 93.23 for r6-k1 and r6-k2 cases respectively when we move from a controlled to an open environment. The overall accuracy for predicting the correct label is 93.75% in a controlled environment and 77.94% in an open environment.

6.6 Model Parameters

We use grid search to find the optimal parameters for the training model. It uses the non-linear radial basis kernel and the validation folds and the number of iterations are restricted to 5 and 300 respectively. One fold is held out for validation at each iteration while the rest are used for training. Two model parameters, namely C and gamma are varied and their optimal value is predicted. C value, that decides the weight for the rate of misclassification is varied in the range of 2^{-5} to 2^{5} and gamma, a parameter of the radial basis kernel is varied from 2^{-4} to 1.

7 Conclusion and Future Work

This paper presents a detailed study of genitive data in the Hindi Treebank. Occurrence of genitives in varied syntactic context is a unique feature of Indo-Aryan languages. We examined the Hindi dependency Treebank and noted down various syntactico-semantic relations in which a genitive modifier occurs. We observed that relations vary from a simple syntactic label r6 to deeper semantic labels-k1, k2. We have attempted to trace syntactic contexts which can be used for predicting the relations automatically. The motivation is to automate the process of labeling genitive data. We have implemented two systems, a rule based system and a statistical system for automatically identifying the attachment of genitive marked noun with its head and the label between them. The statistical model uses the single candidate approach and outperforms the rule based system for the non-contiguous data. The statistical system produces an overall accuracy of 97% in contrast to the rule based system that gives an 95% accuracy for the correct attachment of the genitive. Both the systems perform better than the baseline system presented in Table 2. The output can be verified by the human annotators thus making the Treebank development semi-automatic for the genitive data. Since, it is largely the r6 relation that occurs between two nouns and since for other relations also, the syntactic contexts to a great extent can be identified, the task of automated labeling of genitive data appears very promising in the context of dependency Treebank development.

As a part of the future work, we will integrate our system with the MALT parser or MST parser. The genitive parsing module can be used over the MALT/MST parser output as a post-processing module. This would be a promising attempt for improving the parsing accuracy of genitives in Hindi.

References

Bharati A., Chaitanya V. and Sangal. R. (1995). Natural Language Processing: A Paninian Perspective, Prentice-Hall of India, New Delhi, pp. 65-106.

Bharati A., Husain S., Ambati B., Jain S., Sharma D. and Sangal R.. (2008). Two Semantic features make all the difference in Parsing accuracy. In Proceedings of ICON-08.

Bharati A., Sharma D., Husain S., Bai L., Begam R. and Sangal R. (2009a). AnnCorra: TreeBanks for Indian Languages, Guidelines for Annotating Hindi TreeBank (version – 2.0).

Bharati A., Husain S., Sharma D., Sangal R. (2009b). Two stage constraint based hybrid approach to free word order language dependency parsing, Proceedings of the 11th International Conference on Parsing Technologies, October 07-09, 2009, Paris, France

Bhatt R., B. Narasimhan, M. Palmer, O. Rambow, D. M. Sharma and F. Xia. (2009). Multi-Representational and Multi-Layered Treebank for Hindi/Urdu. In Proc. of the Third Linguistic Annotation Workshop at 47th ACL and 4th IJCNLP.

Chang, C., & Lin, C. (2001). LIBSVM: a library for support vector machines. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm, detailed documentation (algorithms, formulae etc) can be found in http://www.csie.ntu.edu.tw/~cjlin/papers/libsvm.ps.gz.

Ge N., Hale J., and Charniak E. (1998). A statistical approach to anaphora resolution. In Proceedings of the 6th Workshop on Very Large Corpora, pages 161–171, Montreal, Quebec, Canada

Girju R. (2008) Tutorial on semantic relation extraction and its applications. Proceedings of the European Summer School in Logic, Language and Information (ESSLLI), Freie und Hansestadt Hamburg, Germany

Gupta M., Yadav V., Husain S. and Sharma D. (2008). A Rule Based Approach for Automatic Annotation of a Hindi Treebank. In Proc. Of the 6th International Conference on Natural Language Processing (ICON-08), CDAC Pune, India.

Kosaraju P., Kesidi S. R., Ainavolu V. B. R. and Kukkadapu P. (2010). Experiments on Indian Language Dependency Parsing. In proceedings of ICON10 Tool Contest.

Lim, J.-H., Park, S.-Y., Kwak, Y.-J., & Rim, H.-C. (2004). A semi-automatic tree annotating workbench for building a Korean treebank. Lecture Note in Computer Science, 2945, 253–257

McDonald R., F. Pereira, K. Ribarov, and J. Hajic. (2005). Non-projective dependency parsing using spanning tree algorithms. Proceedings of HLT/EMNLP.

Nivre J., J. Hall, J. Nilsson, A. Chanev, G. Eryigit, S. Kübler, S. Marinov and E Marsi. (2007). MaltParser: A language-independent system for data-driven dependency parsing. NLE.

Rosario B., and Hearst M. (2001). Classifying the Semantic Relations in Noun Compounds via a Domain-Specific Lexical Hierarchy, Proceedings of the 2001 Conference on Empirical Methods in Natural Language Processing (EMNLP-01),Pp. 82-90

Surtani N. and Paul S. (2012). Genitives in Hindi Treebank: An attempt for Automatic annotation, In Proceedings of 11th workshop on Treebanking and Linguistic Theories, Lisbon, Portugal

Vapnik V., (1995). The nature of statistical learning theory, Springer-Verlag New York, Inc., New York, NY, 1995

Yang X., Su J. and Tan C. (2005). A Twin-Candidates Model for Coreference Resolution with Non-Anaphoric Identification Capability. In Proceedings of IJCNLP-2005. Pp. 719--730, 2005