A rule based approach for automatic clause boundary detection and classification in Hindi

Rahul Sharma, Soma Paul

LTRC, IIIT Hyderabad

August 12, 2014
Clauses in Hindi

Why Clause?
Clause definition
Clause Classification
Types of Clause

Automatic Clause identification and classification
Related Work
Methodology

Error Analysis

Conclusion and future work
Why Clause?

- The performance of many NLP systems like Machine Translation, Parallel corpora alignment, Information Extraction, Syntactic parsing, automatic summarization and speech applications etc. improves by introducing clause boundaries in a sentence (e.g., Ejerhed, 1988; Abney, 1990; Leff, 1998; Papageorgiou, 1997; Gadde et al., 2010).

- “It is impossible, to process a complex sentence if its clauses are not properly identified and classified according to their syntactic function in the sentence” (Leff, 1998).
What is clause?

- A clause is a grammatical unit that includes, at minimum, a predicate and an explicit or implied subject, and expresses a proposition. (Loos, Anderson, Day, Jordan & Wingate, 1999)

- A clause is a sentence-like construction contained within a sentence. (Sidney Greenbaum and Gerald Nelson, An Introduction to English Grammar, 3rd ed. Pearson, 2009)

- Basic criterion for defining a clause is the presence of a verb phrase. (Leffa, 1998)
What is clause?

We take clause as a grammatical unit which contains,

- a verb or a verb group (main verb and its auxiliaries),
- Explicit or implicit arguments of the verb.

Example: **raam ne khana khaya** or **so gayaa**.
Types of clause based on verb’s type

- **Finite Clause**: Clause that contains a finite verb (verb group).
  - Example: *raam ne khana khaya*
Types of clause based on verb’s type

- **Finite Clause**: Clause that contains a finite verb (verb group).
  - Example: `raam ne khana khaya`

- **Non-finite Clause**: Clause that contains a Non-finite verb (verb group).
  - `raam khaanaa khaakar soyaa`. 
Clauses in Complex/Compound sentences

- Superordinate Clause: Main clause or independent clause in a complex sentence.
- Subordinate Clause: Dependent clause in a complex sentence.
Clauses in Complex/Compound sentences

- Superordinate Clause: Main clause or independent clause in a complex sentence.
- Subordinate Clause: Dependent clause in a complex sentence.
- Example: karan jo mere bhai ka dost hE caalaak hE.
  - Superordinate Clause: karan caalaak hE.
  - Subordinate Clause: jo mere bhai ka dost hE.
Types of Clause

- **Relative Clause**: Clause that modifies noun/pronouns. (Hindi, Yamuna Kachru)
  - Example: kshitij jo khel rahaa tha ghar gayaa.

- **Adverbial Clause**: Subordinate clauses denoting time, place, manner, direction, etc. (Hindi, Yamuna Kachru)
  - Example: jaisa vo karegi vaisa tum karoge.

- **Complement Clause**: Clauses which are introduced by the subordinator/complementizer ki and follow the main clause verb.
  - Example: riyaz ne kahaa ki vo khel rahaa tha.

- **Non-finite Clause**: Clause that contains a Non-finite verb (verb group).
  - Example: Ram khaanaa khaakar soyaa.
Types of Clause

- Relative Clause: Clause that modifies noun/pronouns. (Hindi, Yamuna Kachru)
  - Example: kshitij **jo khel raha tha** ghar gayaa.
Types of Clause

- Relative Clause: Clause that modifies noun/pronouns. (Hindi, Yamuna Kachru)
  - Example: kshitij \textit{jo khel raha\text{a} th\text{a}} ghar gayaa.

- Adverbial Clause: Subordinate clauses denoting time, place, manner, direction, etc. (Hindi, Yamuna Kachru)
  - Example: \textit{jaisa vo karegi} vaisa tum karoge.
Types of Clause

- **Relative Clause**: Clause that modifies noun/pronouns. (Hindi, Yamuna Kachru)
  - Example: kshitij jo khel rahaa tha ghar gayaa.

- **Adverbial Clause**: Subordinate clauses denoting time, place, manner, direction, etc. (Hindi, Yamuna Kachru)
  - Example: jaisa vo karegi vaisa tum karoge.

- **Complement Clause**: clauses which are introduced by the subordinator / complementizer ki and follow the main clause verb.
  - Example: riyaz ne kahaa ki vo khel rahaa tha
Types of Clause

- Relative Clause: Clause that modifies noun/pronouns. (Hindi, Yamuna Kachru)
  - Example: kshitij jo khel raha tha ghar gayaa.
- Adverbial Clause: Subordinate clauses denoting time, place, manner, direction, etc. (Hindi, Yamuna Kachru)
  - Example: jaisa vo karegi vaisa tum karoge.
- Complement Clause: clauses which are introduced by the subordinator / complementizer ki and follow the main clause verb.
  - Example: riyaz ne kahaa ki vo khel raha tha
- Non-finite Clause: Clause that contains a Non-finite verb(verb group).
  - Example: Ram khaanaa khaakar soyaa.
Types of Clause

- Coordinate Clause: Independent clauses in a sentence, joined by conjunctions (or, kyunki etc.)..
  - urmi ne quiz-up khela or rishabh ne cricket dekha.
Automatic Clause identification and classification

- Identification of clauses which contain verbal forms (finite verbs and participles) and classifying them into predefined clause types.
Automatic Clause identification and classification

- Identification of clauses which contain verbal forms (finite verbs and participles) and classifying them into predefined clause types.
- Different Techniques used for the task in past,
  - Rule based approach
  - Statistical approach
  - Hybrid approach
Related Work

- Ram and Devi (2008) have built a Conditional Random Fields (CRFs) based hybrid system for detecting clause boundaries in a sentence. After identifying the clause boundaries they run an error analyzer module to find false boundary markings, which are then corrected by the rule based system, built using linguistic clues.
- (Ghosh et al., 2010) is another rule based system for clause boundary identification for Bengali, where they use statistical approach for clause classification and dependency relations between verb and its argument to find clause boundaries.
Related Work

- Sharma et al. (2013) showed how implicit clause information present in dependency trees can be used to extract clauses in sentences.
- Gadde et al. (2010) reported improvement in parser performance by introducing automatic clause information in a sentence for Hindi in Improving data driven dependency parsing using clausal information.
In this Data flow of our system, E represents number of ‘clause end position(CEP)’ and S represents number of ‘clause start position(CSP)’ marked by our system.

Figure 1: Data Flow
In this module, input sentences are processed and each lexical item is assigned a POS tag, and chunk information. For example:

Input sentence:

(1) raam soyaa.
   Ram sleep+past
   ‘Ram slept.’

Output:

1 (( NP
   1.1 raam NNP
   ))
2 (( VGF
   2.1 soyaa VM
   2.2 . SYM
   ))

--Here ‘NP’ and ‘VGF’ are the chunk tags, and POS tags ‘NNP’ and ‘VM’ stand for Noun and Verb respectively (Bharati et al., 2007; Bharati et al., 2009).
CEP Identification

Hindi mainly being SOV, the verb is taken to be the end of the clause. In cases where a sentence does not end with a verb, the end of sentence is taken as end of the clause. This helps to handle instances of scrambling and ellipses. For example:

(2) siitaa ghar jaa rahii hai aur giitaa bhii.
   Sita home go+present+cont and Gita also
   ‘Sita is going home and so does Gita.’

In example (2), there is an ellipses of the verb ‘jaa rahii hai’ in the second clause ‘giitaa bhii’. In cases like this, our system marks the verb as end of the first clause and sentence end as end of the second clause. The marked boundaries in the sentence after this module will be: ‘siitaa ghar jaa rahii hai ) aur gitaa bhii )’.
CSP Identification for finite clause

This module uses linguistic cues such as relative markers (*jo* ‘that/who’, *jisane* ‘who’), coordinating conjuncts (*aur* ‘and’, *lekin* ‘but’) and so on, to identify the start of clauses. It may be noted that the immediate context of cues is also taken into account at times. For example:

(3) **raam** **jo** **khel** **rahaa** **thaa** **nahii** **aayaa.**
Ram who play+past+conti. not come+present
‘Ram who was playing did not come.’

In example (3), first our module identifies ‘*jo*’ relative marker and marks it as a start of the clause ‘*jo* khel **rahaa** **thaa**’, and then, marks the beginning of the sentence as the start of the other clause ‘*raam* ***nahii*** **aayaa**’. After this, the boundaries marked in example (2) will be : ( **raam** **( jo** **khel** **rahaa** **thaa** **)** **nahii** **aayaa.** )
CSP Identification for non-finite clause

- for Non-finite Clause,
  - Templates/regular expression were prepared.
  - any pattern in sentence matching those templates is considered to be a non-finite clause.
CSP Identification for non-finite clause

- for Non-finite Clause,
  - Templates/regular expression were prepared.
  - any pattern in sentence matching those templates is considered to be a non-finite clause.

- Example: (vaha ( jo khel rahaa tha ) ( ghar jaakara ) soyaa )
  - TEMPLATE used: <Verbgroup> <NP(with null posposition)> <non-finite verb>.
    - if string in sentence matches above template then make <NP> and <non-finite verb> as one non-finite clause.
In case the number of CSPs is not equal to the number of CEPs in a sentence, the Sanity Checker module comes into play. It iterates through the CSP identifier’s output for the sentence and marks the omitted CSPs. For example:

(4) raam ghar gayaa, shyam nahii gayaa.  
Ram home go+past, Shyam not go+past.  
‘Ram went home, Shyam did not go.’

The absence of a coordinator between the two clauses ‘raam ghar gayaa’ and ‘shyaam nahii gayaa’, in Example (4) can lead to potential error of omission of the CSP for the second clause ‘shyaam nahii gayaa’. The output of such a sentence would be:  
‘(raam ghar gayaa) shyam nahii gayaa.)’  
On detecting such an error, the sanity checker would iterate the sentence and mark the omitted CSP, and the output would then be:

‘(raam ghar gayaa) (shyaam nahii gayaa.)’
This module executes, and identifies ‘ki’ complementizer and its clause in the sentence, and modifies the CEP of its parent clause. Example (16) explains this further.

(5) raam ne kahaa ki tum ghar jaao
ram+erg say+past that you home go
‘Ram said that you go home.’

The input for the sentence ‘raam ne kahaa ki tum ghar jaao’ that this module receives would be:
‘(raam ne kahaa) (ki tum ghar jaao)’

The ‘ki’ complementizer module iterates this input and identifies the ‘ki’ complement clause and its CEP. It then modifies this input by moving the CEP, immediate before ‘ki’ complementizer to the position immediate after the CEP of ‘ki’ complement clause. The modified sentence will be:
‘(raam ne kahaa (ki tum ghar jaao) )’
Coordination handler

This module handles embedded coordinated clauses in complex sentence where they fall within the scope of a complementizer, a relative marker or an adverbial marker. It makes a new CSP for these clauses immediately before the complementizer, relative marker or adverbial marker and a new CEP after the CEP of the last embedded coordinate clause. For example:

(6) raam jisne khaanaa khaayaa aur khel khelaa ghar gayaa
    Ram who+rel. food eat+past and game play+past home go+past
    ‘Ram who ate food and played a game, went home.’

Given the output for the example (6), this module identifies the ‘jisne’ the relative marker and inserts a new CSP immediately before it. It also inserts the CEP for their coordinate clauses after the CEP of the last embedded coordinate clause ‘khel khelaa’. The output would be:

(raam ( (jisne khaanaa khaayaa) aur (khel khelaa) ) ghar gayaa.)
Evaluation

- For evaluation, a clause is considered rightly identified iff it’s both end and start are identified correctly.

- Results,
  - Results for Finite clause,
    - Precision: **91.30%**
    - Recall: **91.78%**
  - Results for Non-finite clause: 63 non-finite clauses were correctly identified out of 104.
Automatic Clause Classification

- classify a clause into main clause, complement clause, adverbial clause, relative clause, coordinate clause and non-finite clause.

- uses relative pronouns, corelative pronouns, conjuncts, subordinator for classification.

- Results:

<table>
<thead>
<tr>
<th>Clause Type</th>
<th>Precision%</th>
<th>Recall%</th>
<th>F1 score%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Main Clause</td>
<td>77.90</td>
<td>91.78</td>
<td>84.27</td>
</tr>
<tr>
<td>Coordinate Clause</td>
<td>80.00</td>
<td>70.58</td>
<td>74.99</td>
</tr>
<tr>
<td>Complement Clause</td>
<td>92.30</td>
<td>92.30</td>
<td>92.30</td>
</tr>
<tr>
<td>Relative Clause</td>
<td>93.33</td>
<td>66.66</td>
<td>77.77</td>
</tr>
<tr>
<td>Adverbial Clause</td>
<td>100</td>
<td>50</td>
<td>66.66</td>
</tr>
</tbody>
</table>
Error Analysis

- POS and chunking errors.
  - ‘mill’, name of a girl was given VM as POS tag.
  - ‘lenovo bhi’, chunked as VGNF instead of NP.
- Ellipses of verb in a sentence.

\[(7) \text{raam ne kitaab } \langle V \rangle \text{ aur maine kavitaa padhii} \]
\[
\text{Ram+erg book } \langle \text{read+past} \rangle \text{ and I+erg poem } \langle \text{read+past} \rangle
\]

‘Ram read a book and I read a poem’

- Scrambling in the usual word order, which is SOV in Hindi.

\[(8) \text{ghar gayaa raam, vaha bolaa.} \]
\[
\text{home go+past Ram, he say+past}
\]

‘He said Ram went home’
Error Analysis

- Missing subordinate conjunction ‘ki’ in a sentence also leads to incorrect identification of clause boundaries by our system. For example:

(9)  raam ne kahaa  tum ghar jaao
    Ram+erg say+past you home go
    ‘Ram said you go home’

- Start of non-finite clause: As we don’t find any syntactic cues for start of non-finite clause, templates does not perform well.
We have discussed our work on clause boundary identification and classification in Hindi and the issues pertaining to them. System developed showed satisfactory performance for finite clauses in terms of F1 scores of 91.53% for clause boundary identification and 80.63% for clause Classification, while giving inadequate results for non-finite clauses with 60.57% accuracy. This task is a promising resource for NLP systems such as Machine Translation, Text-to-Speech and so on, and can contribute to their better performance, applying it in those NLP systems seems quite a favorable prospect as a future work.
Any Questions/Suggestions?
THANK YOU.