Automatic identification of Hindi complex predicates for computer-aided reading tool

Satenik Mkhitaryan

Automatic identification of Hindi complex predicates for computer-aided reading tool

Satenik Mkhitaryan
National Institute of Oriental Languages and Civilizations (Inalco)

Abstract

This study demonstrates how Natural Language Processing resources can be deployed in Computer Assisted Language Learning for Hindi as foreign language learners. Many researchers have shown the importance of bringing attention and awareness on language categories and forms in second language learning. We introduce a web based implementation that provides visual enhancement of texts in order to make language learning targets more salient for the learner. Learners get to choose a text they want to read and the system displays an enhanced version of the text. It supports visual enhancement for some simple categories (nouns, adjectives, verbs etc.) and provides the lemma if asked, using a Hindi Part-Of-Speech tagger. It also detects the complex predicates (CP).
Hence, in this study we will focus on automatic identification of Hindi CPs which are known to be problematic for Hindi learners. We assume that highlighting the CPs will help the reader to grasp them as a single unit of verb and not each element of the CP separately, thus contributing to its comprehension and facilitating its acquisition.
We will first review the existing methods for detecting Hindi complex predicates which are often qualified as « pain in the neck for NLP ». We will then present a short overview of the tools and resources for processing Hindi. Finally, we will describe our method for detecting CPs in the context of the abovementioned reading tool and we will discuss the results.

Motivation

Computer Assisted Language Learning (CALL) is a prolific field of computer use for learning and there are many applications of NLP techniques in CALL, called Intelligent CALL (ICALL). Numerous studies on meaning-focused communicative approaches show that input alone is not enough to learn a foreign language. Recognizing the importance of bringing a focus on form in foreign-language learning, many reading tools have been created (GLOSSER[1][12], ALPHEIOS[2], WERTi [10], Didalect [7], NaviLire [8], ARET [9], REAP[3]), most of them for European languages.
AideMoi is a project by ER-TIM (Inalco) which is currently under development. It aims to develop a web based reading tool for language learning. It promotes early and intensive reading practice, automatically enriched with linguistic information which allows the reader to practice independent reading and develop their reading strategies and comprehension, their metalinguistic reflection. The application proposes simple and intuitive functionalities of textual analysis and annotation (highlighting certain parts of speech, showing concordance of lexical units, highlighting certain structures considered as difficult for learners, measuring readability etc.).
In this paper we'll focus on automatic detection of Hindi complex predicates (CP) in AideMoi. A CP is a compound expression combining a verb with a noun, an adjective, an adverb or another verb, but which behaves as a single verb. This structure is often compared to phrasal verbs and to support verb constructions in English. According to (Mukerjee et al. 2006) “Identifying CPs in text

[1] http://www.let.rug.nl/glosser/Glosser/
[2] http://alpheios.net
[3] http://reap.cs.cmu.edu

is crucial to processing since it serves as a clausal head, and other elements in the phrase are licensed by the complex as a whole and not by the verbal head."

Moreover, (Yasuda 2010) has also shown that annotation and enhancement of English phrasal verbs in a text facilitates learning and, compared to a non-annotated text, significantly improves the results of learners.
Thus, we will first review the existing methods for detecting Hindi complex predicates. We will then present a short overview of the tools and resources for processing Hindi. Finally, we will describe our method for detecting CPs in the context of the aforementioned reading tool and we will discuss the results and possible avenues for improvement.

Related Work

Many researchers have studied the issue of automatic detection of complex units in several languages in order to perform various processing tasks. They have adopted different approaches: some use parallel corpora, others use monolingual corpora, some apply statistical methods, others combine them with rules. We will here present the major works on the detection of Hindi complex predicates.

(Mukerjee et al. 2006) used Hindi-English parallel corpora and the Part-Of-Speech (POS) tagging of the English corpus. The considered CP types are Adj+V, N+V, Adv+V and V+V. The authors claim that rule-based approaches are not very effective since there is no computationally implementable rule that can distinguish the CPs from similar but non-CP constructions (ānumati denā [permission+to give] and kitāb denā [book+to give]). On the other hand, CPs may be translated as a simple verb in other languages and POS projection in a parallel corpora may help to detect them. In this method, a checklist of light verbs is also used to decide whether the identified sequence is a CP. In this way, if the projected tag of a Hindi word is Verb and the normal POS tag of the word in the Hindi dictionary is N, Adj, V or Adv and the word is followed by one of the verbs from the light verb list, then the detected multi word expression is classified as N+V, Adj+V, V+V, or Adv+V CP respectively.
(Mukerjee et al.) also discuss the case of discontinuous CPs since their constituents may sometimes be quite far apart in the sentence. They believe that this approach can capture some discontinuous CPs if the source language tags are considered too.
The limitation of this method is that it cannot detect the CPs that have been translated as a CP in English as well. For example, जवाब दे javāb de will be translated into English as "answer" or "give answer." In the second case, the CP will not be detected. While this approach misses some CPs, those that are identified are seen to be quite reliable. Indeed, they report a precision of 83 % and a recall[4] of 46 %.

(Sinha 2009) attempted to identify CPs of all kinds, also using Hindi-English corpora, but unlike (Mukerjee et al. 2006), they project the meaning of the light verb in the parallel corpora presuming that the meaning of the CP is different from the meaning of the light verb.
This system consists of following steps: align the sentences of the parallel corpora; create a list of Hindi light verbs and their translations in English; generate all the morphological forms of Hindi light verbs and of their corresponding English verbs; search for CPs in each Hindi-English aligned sentence. They also use "stop words" (words that can appear between the components of a CP)

[4] Precision and recall are measures used in evaluating search strategies. The precision is the ratio of relevant instances retrieved to the total number of irrelevant and relevant instances retrieved. The recall is the ratio of relevant instance retrieved to the total number of relevant instances in the database.

allowing words within the CPs to improve the performance and "exit words" (words which cannot be part of a CP) to avoid incorrect identification.
This simple method yields an F-score[5] between 88 % and 97 %. These results are much higher than the results of (Mukerjee et al. 2006) whose method would fail to detect CPs such as सलाह लेना salāh lenā [to seek advice] and खुशी होना khuśī honā [to feel happy] which are perfectly identified in this method.

(Chakrabarti et al. 2008) focus on V+V type of CPs, using a Hindi corpus. In this type of constructions, only a certain sub-group is really a CP, such as: inf+paḍnā, inf-e+lagnā, stemV1+V2. The first two constructions are regular and predictable, but in the third type, the choice of the second verb (vector verb[6]) is unpredictable. Authors call them lexical compound verbs and suggest including them in a lexical database by extracting them automatically. Their algorithm is simple: if a verb appears in his base form and is followed by one of the verbs that are in a predefined list, then the two verbs form a lexical compound verb. After this stage, ten native speakers are asked to make sentences with the extracted sequences. If they are able to do so, the sequences are registered as lexical compound verbs. They reach an accuracy of 98 %, but they do not report any information about the recall.

(Begum et al. 2011) tried to identify the conjunct verbs only, using Hindi corpus and a statistical tool. At first, they do some tests to decide manually whether a N+V sequence is a CP or not (they use some of the conditions mentioned in (Bhattacharya et al. 2006): coordination test; constituent response test (Wh-questions); relativization; adding the accusative case marker; adding the demonstrative pronoun). Then, the authors define 7 features which are used in a statistical tool (maximum entropy[7]) for binary classification of a N/Adj+V expression into conjunct verb and non-conjunct verb. The 7 features are : verbs (some verbs are more likely to occur as a light verb), object (some objects have high chances to occur with a light verb), semantic category of objects (ex. "Artifact", "Abstraction", "State", "Physical Object" etc.), post-position indicator (which indicates whether a noun or an adjective is followed by a postposition), demonstrative indicator (which indicates the presence of a demonstrative), frequency of verbs corresponding to a particular object (if a noun / adjective often appears with a particular verb, it is likely that this pair forms a complex predicate), verb argument indicator (average number of postpositions that appear before a N / Adj+V sequence). They use two annotated corpora of Hyderabad Dependency Treebank (4500 sentences for the training corpus and 1800 sentences for the test corpus). They report 85.28 % of accuracy.

NLP tools and resources for Hindi

Automatic processing of Hindi and Indian languages in general is booming in many institutions all over India. The research centers that seem most active in this field are Indian Institut of Technology Bombay[8] (P.Bhattacharya, D.Chakrabarti, V.M.Sharma), International Institute of Information Technology Hyderabad[9] (Rajeev Sangal, Dipti Misra Sharma), Indian Language Technology Proliferation and Deployment Center[10], International Institute of Information Technology Kharagpur and International Institute of Information Technology Kanpur.

---

5 F-score is the harmonic mean of precision and recall.
6 Different terms can be found in the literature to designate the second verb
7 Maximum entropy toolkit http://homepages.inf.ed.ac.uk/lzhang10/maxent_toolkit.html
8 IIT Bombay http://www.cfilt.iitb.ac.in/
9 IIIT Hyderabad http://web2py.iiit.ac.in/research_centres/default/view_area/3
10 ILTPDC http://tdil-dc.in/index.php?option=com_vertical&parentid=2

Since India is a country where many languages coexist, several institutions were soon interested in machine translation between Indian languages as well as from English into Indian languages. The approaches are different, based on rules, parallel corpora, lexicons, statistical methods etc.
We can reference a few projects (AnglaMT, Anuvadaksh, Sampark) that have been realized through the collaboration of fifteen institutions from different regions of India (Hyderabad, Pune, Mumbai, Chennai, Kharagpur, Allahabad, Tamil, Bangalore, Jadavpur etc.) and are funded by the Indian government as part of TDIL project (Technology Development in Indian Languages). Many other systems exist (Mantra, AnglaBharti, Anubharti, Anuvaadak, Hinglish, Anubaad, Shakti, Shiva etc.). For details on machine translation in Indian languages, one can refer to (Garje et al. 2013), (Bandyopadhyay 2004), (Bharati et al. 2000).

Several Part Of Speech (POS) taggers have been created adopting different approaches, based on rules or on statistical methods (Hidden Markov Model, decision trees, CRF, maximum entropy etc.) combined with rules. But, as far as we know, only two taggers are available for free download.

The first one, developed by IIT Bombay researchers, is based on CRF++[11]. The tagger returns only the POS tag of the tokens. The authors use tags defined by A. Bharati[12].

Below are examples of some correctly as well as incorrectly tagged words:

Correct	Incorrect
दिल्ली (Delhi) NNP	मेट्रो (metro) NNP
की (GEN) PSP	किया। (do.PERF.M.SG) NNP
सफ़लता (success) NN	हुई। (be. PERF.F.SG) NNP
हैं। (be.PRES.3.PL) VAUX	चौबीस (twenty-four) NNP
चार (four) QC	है। (be.PRES.3.SG) JJ

Table 1 NNP: proper nouns, PSP: postposition, NN: nouns, VAUX: verb auxiliary, QC: cardinals, JJ: adjective

Another tagger written in Python (based on TnT tagger[13]) has been developed by Siva Reddy from the University of Edinburgh. The author uses the same tags as in the previous tagger, i.e. those of A. Bharati.

This tagger not only provides the POS tag but also some morpho-syntactic information such as lemma, suffix, gender, number, person, case.

Here is an example of the output (the one in gray is incorrect).

Token	Lemma	POS tag	Suffix	Coarse POS tag	Gender	Number	Person	Case
चलते	चल	VM	ता	v	m	pl	any	
रिपोर्टर	रिपोर्टर	NN	0	n	m	sg	3	d
नज़ारा	नज़ारा	VM		n	m	sg		d

Table 2 VM: verb main, NN: noun

[11] CRF (Conditional random fields) is a probabilistic framework used for structured prediction in pattern recognition and machine learning.

[12] http://ltrc.iiit.ac.in/tr031/posguidelines.pdf

[13] http://www.coli.uni-saarland.de/~thorsten/publications/Brants-ANLP00.pdf

Another significant resource, Hindi Wordnet[14], was created by IIT Bombay researchers on the basis of the English Wordnet. It contains nouns, adjectives, adverbs and verbs. For an input, it provides a set of synonyms, a definition, an example in context and the position in the ontology. Hindi Wordnet also establishes links with various wordnets available for other Indian languages.

Other tools and resources have been created (some of which are available for download): morphological analyzer, transliteration tools, annotated corpora, OCR, speech recognition, text to speech conversion, spell checkers, handwriting recognition, cross-lingual search engine etc.

Detecting Hindi CPs in AideMoi

We saw earlier that the detection of CPs in previous works was done either by using parallel corpora or with statistical methods. These approaches turned out to be unattainable, as we didn't have access to such resources as Hindi-English bi-texts or large annotated Hindi corpus. We also saw that the detection of PC based on rules was effective only for very limited number of CPs.

Hence, we adopted a method similar to that of (Meurers et al. 2010) for the detection of English phrasal verbs in Werti, a reading tool. Their method consists of using a list of phrasal verbs considered as a lexical phenomenon. Unlike English, Hindi is a morphologically rich language. Even with a CP list, their detection reveals many difficulties. Account must be taken of all tenses / aspects / moods and morphological transformations specific to certain verbs and tenses.

Many CPs have been stored in Hindi Wordnet. We have a list of 3711 CPs composed of two or three elements. We also have a list of compound verbs, frequent V+V combinations (422 in total) which we added to our list.

The different spellings are taken into account: words borrowed from Arabic and Persian are present twice, with and without the point below some characters (ख़ुश / खुश करना xush/khush karnā [make happy]; words written with or without ligature (ख़तम / ख़त्म करना xatam / xatm karnā [to complete]; the different pronunciations of borrowings from Arabic or Persian मुक़ाबला / मुक़ाबिला करना muqābala / muqābilā karna [to compete]. Incidentally, we note that there are also borrowings from English (ब्लॉक कर देना blok kar denā [to block], टेक ऑफ करना ṭek of karnā [to take off]).

The first component (first two components for trigrams) is usually unchanged. It may be variable if it is an adjective, noun or a variable participle.

These variations are sometimes present in the list:

नारा_लगाना nārā lagānā	नारे_लगाना nāre lagānā	shout slogans
अनसुना_करना ansunā karnā	अनसुनी_करना ansunī karnā	ignore
पूरा_करना pūrā karnā	पूरी_करना pūrī karnā	accomplish

In the method we present below, changes in the first component are not considered because their processing would need to have a robust lemmatizer which does not yet exist for Hindi.

[14] Wordnet is a lexical database. It groups words into sets of cognitive synonyms which are interlinked by means of semantic and lexical relations.

Thus, our task is to detect, with a regular expression[15] and in an optimal way, CPs in a text by using a CP list, taking into account the following constraints:

1. Morpho-phonological changes in verbal stem
2. Tenses / aspects / moods
3. Voice
4. Particles between components
5. Distance between the components
6. Vector verbs
7. Spellings of verb endings

The morpho-phonological changes in verbal stem are possible in the following cases:

- to form the past participle, imperative, subjunctive (and therefore the future) of some verbs (karnā, lenā, denā, honā, jānā)
- when verbal stem ends in a vowel, the glide -y- is inserted between the stem and the ending

The tenses, aspects, moods and voice are summarized in the following diagram classified according to the type of formation (for convenience for regular expressions). Presumptive being compatible with various aspects, tenses and moods, we add an asterisk when it's the case.
The boxes which may undergo morpho-phonological transformations are framed in black.

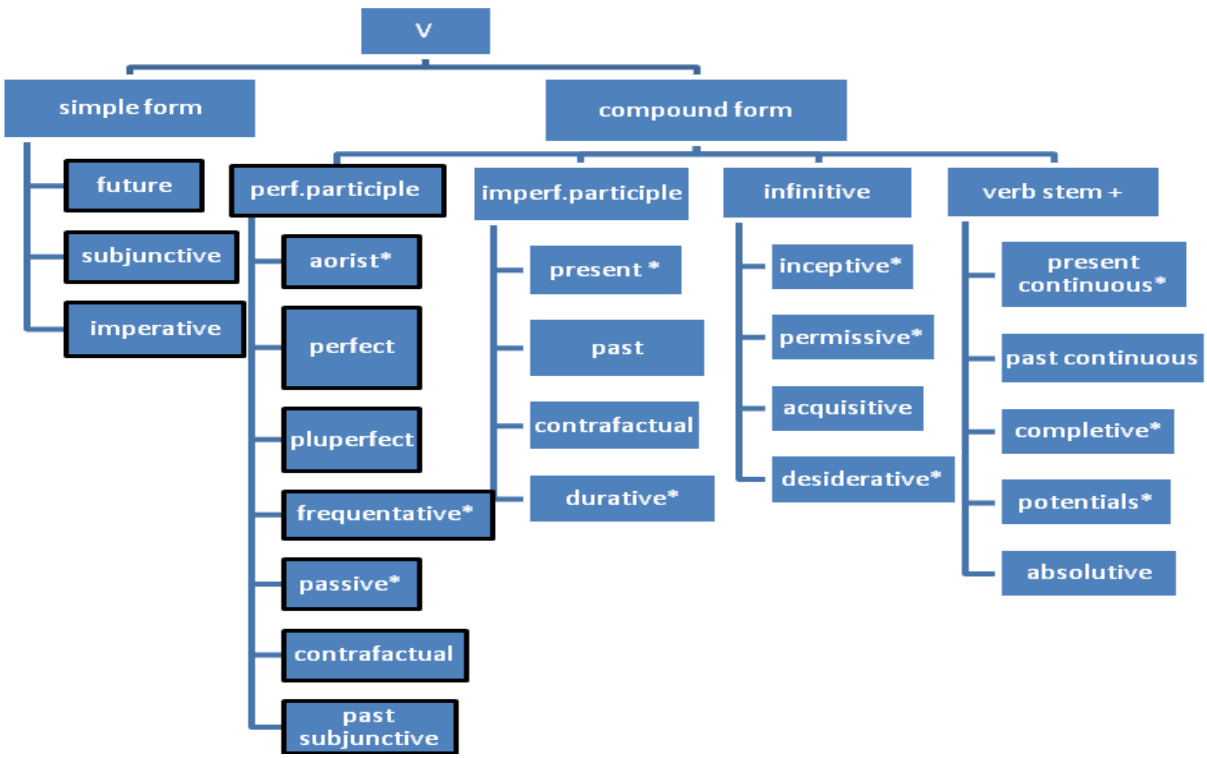


Figure 1 Tenses, aspects, moods, voice

[15] A regular expression (abbreviated regex or regexp and sometimes called a rational expression) is a sequence of characters that define a search pattern, mainly for use in pattern matching with strings. (Wikipedia)

The syntagmatic and paradigmatic variations of Hindi predicate can be summarized in the following scheme:

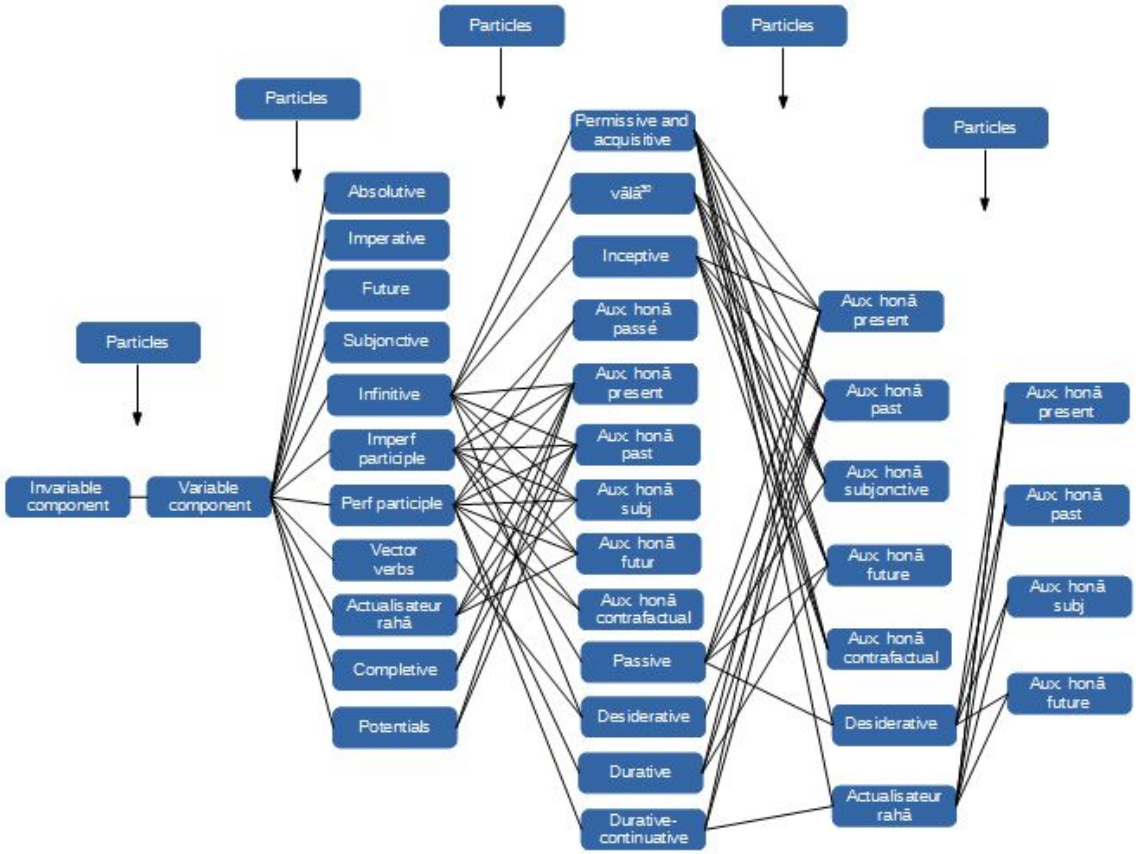


Figure 2 Syntagmatic and paradigmatic variations

Here, we have chosen to represent the most common combinations and not all possible combinations for reasons of readability.

Using the above figures and tables which summarize the formation of different tenses, aspects, moods and voice, we implemented a regular expression that enables the detection of CPs which have regular verbal forms (and a slightly different regular expression for CPs with irregular verbal components).

Experimentation and Results

We tested our method on two different corpora: a newspaper corpus and a corpus of literary texts. The newspaper corpus is composed of NDTV Hindi website's articles published in the year of 2013, it contains 117,083 words. The articles were automatically aspirated and textual content was extracted using Perl scripts. The literary corpus (116,659 words) contains extracts of novels, short stories and plays of Indian authors. The texts were collected from the website of "Mahatma Gandhi International Hindi University".
On the newspaper corpus, a total of 6216 PC were detected while on the literary corpus is a total of 5979 CPs that have been detected.

In order to assess more accurately our results, we manually annotated part of each corpus. We

decided to do two different assessments. The annotated newspaper corpus was used to evaluate the detection of N / Adj / Adv + V type of CPs, while the literary corpus was used to evaluate the V+V combinations.
The annotated newspaper corpus (5354 words) contains a total of 250 complex predicates. The system detects 241 including 9 incorrect.
This gives an accuracy of 96.2%. The errors are due to the syntactic ambiguity of the form kī which can be either a possessive marker or the past-tense form of the verb karnā [to do].

Here are some examples of incorrectly detected sequences:

सेवा	की	टीमों	ने…
Sevā=kī		ṭīmõ=ne…	
Service=GEN		team.OBL.PL=ERG	

लड़की	से	बलात्कार	की	कोशिश	का	मामला…
Ladkī=se		blātkar =kī		kośiś=kā		māmlā…
Girl=SOC		rape=GEN		attempt=GEN		case

The recall is 92.8%. Undetected CPs can be divided into three categories.

1. CPs absent from the list. Note that this category contains a significant number of CPs whose non verbal component is borrowed from English :

अपसेट	हो	गए	थे
Apseṭ	ho	gae	the
Upset	become	go.PERF.M.PL	be.PAST.M.PL

रजिस्टर्ड	कराया	गया	था
Rajisṭard	krāyā	gayā	thā
Registered	do.CAUS	go.PERF.M.SG	be.PAST.M.PL

2. Spelling error in the text (*giftār* instead of *giraftār*):

गिफ्तार	कर	लिया	गया	है
Giftār	kar	liyā	gayā	hai
Arrest	do	take.PERF.M.SG	go.PERF.M.SG	be.PRES.3.SG

3. The distance between the components of the CP (here they are shown in italic)

	अकादमी	की	21	दिसम्बर	को	हुई	बैठक	में
Nir ay	akādamī=kī	21	disambar=ko	huī	baiṭhak=mẽ	*liyā*	*gayā*	
Decision	academy=GEN	21	december=DAT	be.PERF.M.SG	assembly=in	take.PERF.M.SG	go.PERF.M.SG	

	जल्द से जल्द	
*Phaislā*jald se jald	*kiyā*	*jāegā*
Decision as soon as possible	do.PERF.M.SG	go.FUT.M.SG

The annotated literary corpus (5399 words) contains 135 compound verbs. 116 compound verbs were detected including 1 incorrect. The error is due to the morpheme kar which has been identified as the beginning of a compound verb but which in reality was the absolutive:

बुला	कर	ले	गए
Bulā	kar	le	gae
Call	ABS	take	go. PERF.M.PL

This gives an accuracy of 99.1% and a recall of 85.9%. A lower recall was expected given the relatively unpredictable use of vector verbs.

Here are some examples of detected compound verbs:

कर	दी	होती
Kar	dī	hotī
Do	give.PERF.F.SG	be.IMPF.F.SG

छोड़	दूँगा
Chod	dũgā
Let	give.FUT.M.SG

टूट	गई	है
ṭuṭ	gaī	hai
Break	go. PERF.F.SG	be.PRES.3.SG

बचा	ले	जाती	है
Bacā	le	jātī	hai
Save	take	go. IMPF.F.SG	be.PRES.3.SG

रो	पड़े
Ro	padi
Cry	fall.PERF.M.PL

बता	देना	चाहिए	था
Batā	denā	cāhie	thā
Tell	give	need	be.PAST.M.S

Below are some examples of compound verbs that were not detected because they were missing from our list:

उठ	आई	थीं
uṭh	āī	thĩ
Get up	come.PERF.M.PL	be. PAST.F.PL

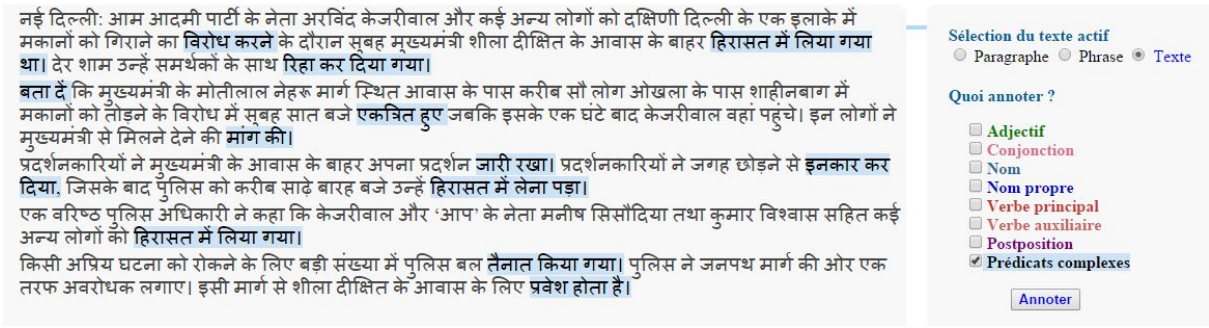
समझ	बैठा	था
Samajhbaiṭhā	thā	
Understand	sit.PERF.M.SG	be.PAST.M.SG

लड़	बैठे

laḍ baiṭhe
Dispute sit.AOR.P

हो उठा
Ho uṭhā
Be get up.PERF.M.PL

Thus, we added the feature of Hindi CPs' detection in AideMoi. The following figure illustrates the result:



In this text, the system detected both compound and conjunct verbs composed of two (virodh karnā [to oppose], inkār karnā [to refuse], batā denā [to tell], etc.) as well as three (hirāsat mẽ lenā [to arrest]) components. The verbs are conjugated in different tenses/aspects/moods and voice.

Conclusion

This work allowed the integration of the detection of complex predicates into a reading tool – AideMoi.

It has been clearly shown that solid linguistic knowledge was prerequisite for the detection and annotation of complex units such as complex predicates. Indeed, our method of using a list of complex predicates requires taking into account the complexity of Hindi verb phrase.

Regarding the effectiveness of the method, evaluation tests on two small corpora reveal high accuracy and lower recall which was expected given the nature of the method. However, the important aspect for CALL is the accuracy. We are aware that we should carry out tests on bigger annotated corpus to better evaluate the method.

The observed results reveal the limits of the method. Thus, we should first emphasize that the list used is not exhaustive and may never be. We also noticed that some complex predicates are not detected even if they are in the list. This is happens in the case of discontinuous CPs, CPs whose components are separated from each other by other words. Finally, there were erroneously detected CPs, alias “false positives”, due to the morphosyntactic ambiguity of some morphemes.

Further to our research, we can now isolate complex predicates. The application for this detection is not limited to our initial motivation, but it can also be used to diversify the functionalities of AideMoi such as setting up a concordance of CPs or assessing the readability of texts (the number of CPs in a text could be a criterion of difficulty).

References

[1] Bandyopadhyay, S., 2004. Use of machine translation in India. AAMT Journal, 36, 25-31.

[2] Begum, R., Jindal, K., Jain, A., Husain, S., Sharma, D.M., 2011. Identification of Conjunct Verbs in Hindi and Its Effect on Parsing Accuracy, in: Gelbukh, A.F. (Ed.), Computational Linguistics and Intelligent Text Processing, Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 29–40.

[3] Bharati, A., Chaitanya, V., & Sangal, R. (2000, October). Computational linguistics in India: an overview. In Proceedings of the 38th Annual Meeting on Association for Computational Linguistics (pp. 1-2). Association for Computational Linguistics.

[4] Bhattacharyya, P., Chakrabarti, D., Sarma, V.M.: Complex predicates in Indian languages and wordnets. In: Language Resources and Evaluation 40(3-4): 331-355 (2006)

[5] Chakrabarti, Debasri, M andalia Hemang, Priya Ritwik, Sarma Vaijayant hi, Bhattacharyya Push-pak. 2008. Hindi Comp ound Verbs and their Automatic Extraction. International Conference on Computational Lingui stics –2008 , pp. 27-30 .

[6] Garje, G. V., & Kharate, G. K., 2013. Survey of Machine Translation Systems in India. International Journal on Natural Language Computing (IJNLC), 2(4), 47-67.

[7] Hermet, M., Szpakowicz, S., Duquette, L., 2006. Automated Analysis of Students' Free-text Answers for Computer-Assisted Assessment. Presented at the The 13th Conference on Natural Language Processing (TALN 2006). April 10-13, 2006. Leuven (Belgium), pp. 835–845.

[8] Lundquist, Lita; Minel, Jean-Luc; Couto, Javier. (2006). NaviLire, Teaching French by Navigatingin Texts. 2006. Conference: The 11th International Conference. IMPU 2006. Information Processing and Management of Uncertainty in Knowledge-based Systems, No. 11, Paris, Les Cordeliers, France, July 2, 2006 - July 7, 2006.

[9] Maamouri, M., Zaghouani, W., Cavalli-sforza, V., Graff, D., Ciul, M., n.d. Developing ARET: An NLP-based Educational Tool Set for Arabic Reading Enhancement.

[10] Meurers, D., Ziai, R., Amaral, L., Boyd, A., Dimitrov, A., Metcalf, V., Ott, N., Tübingen, U., n.d. 2010. Enhancing Authentic Web Pages for Language Learners.

[11] Mukerjee, A., Soni, A., Raina, A.M., 2006. Detecting Complex Predicates in Hindi Using POS Projection Across Parallel Corpora, in: Proceedings of the Workshop on Multiword Expressions: Identifying and Exploiting Underlying Properties, MWE '06. Association for Computational Linguistics, Stroudsburg, PA, USA, pp. 28–35.

[12] Nerbonne, J., Paskaleva, E., Karttunen, L., Proszeky, G., Roosmaa, T., 1997. Reading more into foreign languages, in: Proceedings of the Fifth Conference on Applied Natural Language Processing. Association for Computational Linguistics, pp. 135–138.

[13] Sinha, R.M.K., 2009. Mining Complex Predicates in Hindi Using a Parallel Hindi-English Corpus, in: Proceedings of the Workshop on Multiword Expressions: Identification, Interpretation, Disambiguation and Applications, MWE '09. Association for Computational Linguistics, Stroudsburg, PA, USA, pp. 40–46.

[14] Yasuda, S., 2010. Learning Phrasal Verbs Through Conceptual Metaphors: A Case of Japanese EFL Learners. TESOL Quarterly 44, 250–273. doi:10.5054/tq.2010.219945