# Repairing Bengali Verb Chunks for Improved Bengali to Hindi Machine Translation

*Sanjay Chatterji, Nabanita Datta, Arnab Dhar, Biswanath Barik, Sudeshna Sarkar, Anupam Basu*
Department of Computer Sc. & Engineering, Indian Institute of Technology, Kharagpur, India
Email:schatt@cse.iitkgp.ernet.in,nabanita.2121@gmail.com,{arnabdhar,bbarik,sudeshna,anupam}@cse.iitkgp.ernet.in

ABSTRACT

The present paper identifies the mistakes made by a data driven Bengali chunker. The analysis of a chunk based machine translation output shows that the major classes of errors are generated from the verb chunk identification mistakes. Therefore, based on the analysis of the types of mistakes in the Bengali verb chunk identification we propose some modules. These modules use tables of manually created entries which are validated using chunk annotated and dependency annotated corpus. These modules are used to repair the Bengali verb chunks and subsequently to improve the quality of Bengali to Hindi transfer based machine translation system.

## 1 Introduction

Bengali is a verb ending language (in general). The features of a Bengali finite verb include tense, aspect, mood, person, emphasizer, and voice. Similarly, Hindi is also a verb ending language (in general) and the features of a Hindi finite verb include tense, aspect, mood, gender, number, person, emphasizer, and voice.

In a Bengali to Hindi chunk based machine translation system the chunk tags and chunk boundaries of the Bengali input sentence are identified by a baseline data driven chunker with the help of a model built from a manually annotated training data. We have annotated 9000 Bengali sentences (250K words) manually using a chunk tagset containing 11 tags. This baseline data driven Bengali chunker often makes mistakes in identifying chunk boundaries and chunk tags. These mistakes subsequently give rise to errors in chunk based Bengali to Hindi machine translation system. Specifically, the mistakes in the verb chunks are the source of a major class of translation errors.

This paper works to improve the tags and boundaries of the Bengali verb chunks returned by the data driven chunker in order to improve the machine translation quality. As a preprocessing stage, a large Bengali corpus is annotated by this data driven chunker. We analyze the output of this chunker to formulate the rules to correct the misidentifications of the Bengali verb chunks and created some tables required to implement the rules. This data driven baseline chunker combined with the rule based modification can be considered as a hybrid chunker. We evaluate the performance of both the data driven and hybrid chunkers in terms of precision, recall and f-measure. We also evaluate the applications of the data driven and the hybrid Bengali chunkers in a transfer based Bengali to Hindi machine translation system in terms of BLEU and NIST scores.

*Proceedings of the 10th Workshop on Asian Language Resources*, pages 65–74,
COLING 2012, Mumbai, December 2012.

## 2 Related Work

Pal and Bandyopadhyay (2012) have used aligned English and Bengali chunks to improve the performance of the English-Bengali Phrase Based Statistical Machine Translation System (PB-SMT). They have aligned English chunks with Bengali chunks automatically by validating the translations of English chunks with the original Bengali chunks. Only for the case of verb chunk alignment they were able to compare the translations of English verb chunks with the Bengali verb chunks as verb chunks have one-to-one correspondence in the English and Bengali chunk annotated sentences.

The nature of formation of different kinds of Bengali chunks are described by Das et al. (2005) and implemented by Das and Choudhury (2004). They have used the features of the arguments and predicates of the verbs to identify different chunks.

Due to unavailability of adequate training data, chunking for Bengali language has been attempted in rule based techniques. Bandyopadhyay and Ekbal (2006) have used some handcrafted Bengali specific rules to identify Bengali chunk boundary. These rules help in checking whether two neighboring POS tags belong to same chunk. The chunk tags are assigned based on the POS tags of the member words.

The task of identifying the chunks in Hindi sentences using syntactic rules has been carried out by Bharati et al. (1995) and using rewrite rules by Ray et al. (2003). Vilain and Day (2000) have used the transformation rules in the identification of the chunk tags from the POS tags.

Language specific rules are used to improve the performance of the data driven chunkers in hybrid framework by Bhat and Sharma (2011) for languages like Hindi, Kasmiri, etc. Bengali specific rules can also be used as a postprocessor with the data driven Bengali chunker of Dandapat (2007).

## 3 Our Work

### 3.1 Types of Chunks

As a preprocessing stage we have used a manually chunk annotated 9000 sentences (250k words) Bengali corpus to train a statistical model. This training data contains some top level chunk tags namely, noun chunk, verb chunk, adjectival chunk, adverbial chunk, conjuncts, etc.

We tag four types of Bengali verb chunks.

- Finite Verb Group (VGF) is used to indicate the features (tense, aspect, etc.) of the action of the corresponding clause.
- Non-Finite Verb Group (VGNF) is used to indicate intermediate action of the clause.
- Infinite Verb Group (VGINF) is used to mark infinitival (-te ending) verb forms.
- Gerundial Verb Group (VGNN) is used to mark gerundial (-A ending) verb forms.

These Bengali verb chunks have two parts.

- Main Verb (VM) is the compulsory part and is the main meaning bearing component of the verb chunk.
    - Sometimes, a single (finite or nonfinite) verb is used as VM. In the Bengali VGF বলে ফেলেছি (bale phelechhi) [have told] the nonfinite verb বলে (bale) [tell] acts as VM and in the Bengali VGF বলেছি (balechhi) [told] the finite verb বলেছি (balechhi) [told] acts as VM.
    - Sometimes, a complex predicate is used as VM. চোখে পড়া (chokhe pa.DA) [see] is an example Bengali complex predicate where the noun চোখে (chokhe) [eye] is related to the verb পড়া (pa.DA) [fall] by part-of dependency relation.
- A verb chunk may optionally contain a sequence of Auxiliary Verbs (VAUX) that follow the VM part. This sequence contains a sequence of nonfinite verbs followed by a finite verb. In the Bengali VGF বলে ফেলেছি (bale phelechhi) [have told] the finite verb ফেলেছি (phelechhi) [had] acts as VAUX.

## 3.2 Data Driven Chunker

We have implemented a data driven chunker that takes this 9000 sentence training data to build the model by implementing Conditional Random Field (CRF) of Lafferty et al. (2001) using the following feature set.

1. Word Features: Wi-2, Wi-1, Wi, Wi+1, Wi+2, (Wi-1, Wi), (Wi, Wi+1), (Wi-1, Wi, Wi+1).
2. POS And Chunk Features: POSi-2, POSi-1, POSi, POSi+1, CHUNKi-1.
3. Morphological features: MORi, Di, ABBi, LENGTHi, UNKi

In this feature set 'i' is the current position. The names of the attributes are given below.

- W – The Word
- POS – The Part-Of-Speech of the word
- CHUNK – The chunk tag of the word
- MOR – The morphological features and the suffix of the word
- D –Whether the word is a digit or not
- ABB – Whether the word is an abbreviation or not
- LENGTH – Whether the length of the word (number of characters) is greater than 4 or not
- UNK – Whether the word is an unknown word or not

This model is used to test 200 Bengali sentences. The Precision, Recall and F-measure of this data driven chunker on these test sentences are found to be 93.16, 86.64 and 89.78, respectively.

## 3.3 Analysis of mistakes of the data driven chunker

After analyzing the output of the data driven chunker we found that there are many errors. When we apply this data driven chunker to machine translation we found that some of these mistakes lead to errors in translation. As a post-processing step we are preparing rules to correct the Bengali verb chunks which affect the machine translation.

The types of mistakes in the identification of different types of Bengali verb chunks by this data driven chunker are discussed below. Based on the analysis we have categorized the verb chunk mistakes as follows.

1. Sometimes, the noun part of the complex predicate is kept outside the verb chunk by the data driven chunker. Again, the noun preceded by a verb chunk which is not the part of the complex predicate are kept inside the chunk. These kinds of mistakes are referred to as complex predicate related errors.

   In the Bengali phrase মনে করো এটা ভালো (mane karo eTA bhAlo) [suppose this is good] the complex predicate মনে করো (mane karo) [suppose] is broken into two parts মনে (mane)[in-mind] and করো (karo) [do] by the data driven chunker. Subsequently the Bengali phrase is translated to the wrong Hindi phrase मन में करो यह अच्छा है (mana me.N karo yaha achchhA hai).

2. Sometimes, the data driven chunker breaks a verb chunk into two chunks and sometimes it is added with the surrounding verbs to make a single chunk. These kinds of mistakes are referred to as compound verb related errors.

   In the Bengali phrase সে পেনটা বলে ফেলল (se penaTA bale phelala) [he dropped the pen after saying] the VGF ফেলল (phelala)[dropped] is added with the previous verb বলে (bale)[saying] to make the VGF chunk বলে ফেলল (bale phelala) [said] by the data driven chunker. Subsequently the Bengali phrase is translated to the wrong Hindi phrase बह पेन बोल चूका (baha pena bola chukA).

3. Sometimes, the data driven chunker tags some non-finite verbs in the VGNF chunks as finite verbs and sometimes as postpositions. In Bengali the nonfinite verbs of the form *a*e (bale, kare, dhare, etc.) are also used as finite verb. Similarly, the Bengali postpositions which are derived from verb root are also used as nonfinite verbs. These are referred to as VGNF related errors.

   In the Bengali phrase কথাটা বলে চলে গেল (kathATA bale chale gela) [went after telling the words] the VGNF বলে (bale) [telling] is tagged as VGF by the data driven chunker. Subsequently the Bengali phrase is translated to the wrong Hindi phrase बात कहता है चला गया (bAta kahatA hai chalA gayA).

## 3.4 Handling mistakes of the data driven chunker

Based on observation of mistakes of the Bengali baseline data driven chunker we build the following list based modules to correct them.

### 3.4.1 Module of handling complex predicate related errors

The baseline transfer based Bengali to Hindi machine translation system translates the noun part of the complex predicate separately. We have prepared a list of Bengali complex predicates and their translations in Hindi. This parallel Bengali Hindi complex predicate list is used to solve the complex predicate related errors.

However, some of the complex predicates can be translated by translating noun part and verb part separately. Sometimes, this approach of translation leads to incorrect translations. Some of the Bengali complex predicates which needs to be translated as a whole and their translation in Hindi and English are listed in Table1.

| **Bengali Complex Predicate** | **Hindi Translation** | **English Translation** |
|---|---|---|
| রোপন করা (ropana karA) | रोपना (ropanA) | Transplant |
| আঘাত লাগা (AghAta lAgA) | लगना (laganA) | Embark |
| সেলাই করা (selAi karA) | सिलना (silanA) | Stitch |
| মনে হওয়া (mane haoYA) | लगना (laganA) | Seem |
| খিটখিট করা (khiTakhiTa karA) | टोकना (TokanA) | Punctuate |
| কাজে লাগা (kAje lAgA) | काम आना (kAma AnA) | Inure |

TABLE 1 – Examples of Bengali complex predicates whose word by word translations lead to incorrect translations.

### 3.4.2 Module of handling unique representations

Some (verb, verb) pairs always have unique representations independent of the context. We make lists of such pairs for each type of representation. If the current (verb, verb) sequence exists in a list then the rule says that the sequence is represented in the corresponding way. Some of the examples of Bengali (verb, verb) pairs and their unique representations are listed in Table 2.

| VM VAUX | PSP VM | VM VM |
|---|---|---|
| মরে যাওয়া (mare yAoYA) | দিয়ে হওয়া (diYe haoYA) | গিয়ে দেখা (giYe dekhA) |
| পাওয়া যাওয়া (pAoYA yAoYA) | থেকে চলা (theke chala) | আসতে চাওয়া (Asate chAoYA) |
| আছড়ে পড়া (Achha.De pa.DA) | | আটকে রাখা (ATake rAkhA) |

TABLE 2 – Examples of Bengali (verb, verb) pairs with unique representations

### 3.4.3 Module of handling ambiguous representations

In a sentence when a (verb, verb) pair is represented as (VM, VM), then these two verbs are in two different chunks. In the (VM, VAUX) representation, the pair is in a single chunk. Similarly, if the first verb of the (verb, verb) pair be represented as PSP and the second verb as VM, then these two words are in two different chunks. To resolve these conflicts, we enlist the ambiguous verbs along with the features of their dependents. This list is referred to as the demand frames of the verbs. Demand frame of a verb enlists the features of its dependents in a tabular form.

In this list we have also stored the features of the nouns with whom the verb may be attached as postposition. For instance, the থেকে (theke) postposition may co-occur with the noun which has 'র' (ra) or '০' (Zero) suffix in the singular number. So, an entry in this list is "থেকে র|০" (theke ra|Zero). In a Bengali corpus with 20,000 sentences, there are 103 different postpositions, out of which 11 are generated from verb root. These verb rooted postpositions are also used as non-finite verbs. These postpositions cum verbs are করে (kare), পরে (pare), ধরে (dhare), হয়ে (haYe), ছাড়া (chhA.DA), হতে (hate), নিয়ে (niYe), দিয়ে (diYe), থেকে (theke), ভাবে (bhAbe ) and চেয়ে (cheYe).

This manually created list is validated by finding the entries in the Bengali Treebank. The features of the dependents of the verbs in the list are compared with features of its dependents in the Bengali Treebank. Similarly, the features of the noun with whom a verb may act as postposition as described in the list are compared with features of the noun with whom this verb acted as postposition in the Treebank.

If the current (verb, verb) sequence and the features of the associated dependents exist in the list then the rule says that the sequence belongs to the corresponding representation.

### 3.4.4 Module of identifying misidentification of auxiliary verb POS tag

A list of Bengali auxiliary verbs is created manually. If the second verb of the (verb, verb) sequence is there in that list then this sequence may be used as a single chunk otherwise not. The same list is also used for checking the validity of the first verb as auxiliary verb. Examples of Bengali auxiliary verb roots are হওয়া (haoYA) [to be], চলা (chalA) [go], থাকা (thAkA) [remain], etc.

## 4 Evaluation

We have taken 13199 Bengali sentences for formulating the proposed rules. As a preprocessing stage, we have executed Bengali morphological analyzer and Bengali Part-of-Speech tagger on these sentences. The rules are formulated based on the mistakes in the statistical chunking of these sentences.

## 4.1 Evaluating the performance of chunking

To test the performance of the proposed rules in correcting the chunks we have used 200 Bengali sentences from another distribution. The chunk tags and chunk boundaries

in these test sentences given by the baseline data driven chunker are modified using the proposed modules in a sequential process. Amount of corrections of mistakes by each of these modules are given below.

The list of Bengali complex predicates contains 1063 entries. This list identified 135 complex predicates in these test sentences. Module for handling complex predicate related errors improved the chunking of 32 complex predicates.

The list of Bengali (verb, verb) pairs which have unique representations contains 211 entries. Using this list, module for handling unique representations identified 12 wrongly interpreted (verb, verb) pairs in these test sentences and corrected the respective chunks.

We have used demand frames for 312 Bengali verbs and for 11 Bengali post-positions. The module for handling ambiguous representations has improved the 21 chunks using these demand frames.

The list of 24 Bengali auxiliary verbs has been used in the module for identifying misidentification of auxiliary verbs by the POS tagger to improve 6 chunks.

The number of entries in each list and the number of chunk corrections in the test sentences are shown in Table 3. The performance of the data driven chunker and the hybrid chunker (data driven chunker followed by modules) are shown in Table 4 in terms of precision, recall and f-measure.

| Name of the list | Number of entries | Number of corrections |
|---|---|---|
| Complex Predicate | 1063 | 32 |
| Unique | 211 | 12 |
| Demand Frame | 312+11 | 21 |
| Auxiliary Verb | 24 | 6 |

TABLE 3 – The lists with number of entries and the corresponding number of corrections

| | Precision | Recall | F-measure |
|---|---|---|---|
| Data Driven Chunker | 93.16 | 86.64 | 89.78 |
| Hybrid Chunker | 93.42 | 88.02 | 90.64 |

Table 4 – Performance of data driven and hybrid chunkers

## 4.2 Evaluating the application of chunk modification modules in Machine Translation

Both the data driven chunker and the hybrid chunker are integrated into the Bengali to Hindi transfer based machine translation system. The translation system with the data driven chunker and that with the hybrid chunker show the effect of the proposed modules in the automatic translation of the Bengali sentences to Hindi.

Some of the example sentences with the chunk boundaries and chunk tags assigned by

the baseline chunker and modified by the proposed rule based system are shown below. Their corresponding Hindi translations are also shown. In these examples the chunk tags and boundaries of the Bengali sentences given by both the baseline and hybrid systems are shown in Hindi outputs using Angle Brackets (<>) and Underscores (_), respectively.

1. Bengali Input (BI): এখানে আমরা নকশাখচিত শিবলিঙ্গ দেখলাম, আর ঘুরে বেড়ালাম পুরো মন্দির এলাকা । (ekhAne AmarA nakashAkhachita shibali~Nga dekhalAma, Ara ghure be.DAlAma puro mandira elAkA.)

   Baseline Output (BO): यहाँ हम नकशाखचित शिवलिंग देखे, और <घूमके>_VGNF <घुमा>_VGF पूरा मंदिर इलाका । (yahA.N hama nakashAkhachita shibali.nga dekhe, aura ghumake ghumA purA ma.ndira ilAkA.)

   Modified Output (MO): यहाँ हम नकशाखचित शिवलिंग देखे, और <घूम लिया>_VGF पूरा मंदिर इलाका । (yahA.N hama nakashAkhachita shibali.nga dekhe, aura ghuma liyA purA ma.ndira ilAkA.)

   Analysis: This is corrected using the rule that the word pair (ঘুরে বেড়ানো) should be a complex predicate as discussed in the module for handling complex predicate related errors.

2. BI: আমরা দোলমঞ্চ ঘুরে দেখে রাজবাড়ি থেকে বেরোলাম. (AmarA dolama~ncha ghure dekhe rAjabA.Di theke berolAma.)

   BO: हम दोलमंच <घूमके>_VGNF <देखके>_VGNF राजमहल <रहके निकला>_VGF। (hama dolama.ncha ghumake dekhake rAjamahala rahake nikalA.)

   MO: हम दोलमंच <घूमके>_VGNF <राजमहल से>_NP <निकला>_VGF। (hama dolama.ncha ghumake dekhake rAjamahala se nikalA .)

   Analysis: This is corrected using the rule that the (verb, verb) pair থেকে বেরোলাম (theke berolAma) can only be used as (PSP, VM). So, it is stored in the unique list as discussed in the module for handling unique representations.

3. Bengali Input (BI): অনেক কিছু দেখা হয়ে গেল । (aneka kichhu dekhA haYe gela.)

   Baseline Output (BO): बहत कुछ <देखना >_NP <हो गया>_VGF । (bahata kuchha dekhanA ho gaYA.)

   Modified Output (MO): बहत कुछ <देख लिया>_VGF । (bahata kuchha dekha liyA.)

   Analysis: This is corrected using the rule that the verb দেখা হওয়া (dekhA haoYA) with the karma (object) অনেক কিছু (aneka kichhu) should be considered as a single chunk as discussed in the module for handling ambiguous representations.

4. BI: চমত্কার কারুকার্যময় পুকুরঘাট পেছনে ফেলে দাঁড়িয়ে মন্দিরটি। (chamatkAra kArukAryamaYa pukuraghATa pechhane phele dA.N.DiYe mandiraTi.)

   BO: खूबसूरत कारुकार्यमय तालाब घाट <पीछे>_NP <गिराके खरा है>_VGF मंदिर । (KUbasUrata kArukAryamaya tAlAba ghATa pIchhe girAke kharA hai ma.ndira.)

MO: खूबसूरत कारुकार्यमय तालाब घाट <पीछे छोरके>_VGNF <खरा है>_VGF मंदिर । (KUbasUrata kArukAryamaya tAlAba ghATa pIchhe chhorake kharA hai ma.ndira.)

Analysis: This is corrected using the rule that the verb দাঁড়িয়ে (dA.N.DiYe) can't be used as VAUX. Therefore, they must be in two different chunks. This rule is discussed in the module for identifying misidentification of auxiliary verbs.

The BLEU and NIST scores of the translations of 180 sentences in these baseline and modified MT systems are shown in Table 4.

| | **BLEU** | **NIST** |
|---|---|---|
| Baseline MT system | 0.0988 | 3.3288 |
| Modified MT system | 0.1085 | 3.5087 |

TABLE 5 – BLEU and NIST scores of the Baseline and Modified MT systems

## 5 Conclusion

The chunk correction modules prepared for Bengali language can be adopted for similar other Indian languages, like Hindi. The lists required for these modules are easy to formulate. Whenever a mistake is found in the chunking then this can be resolved instantly by inserting some entry in these lists.

More modules may be developed by observing more data. These enhanced modules may also improve the performance of the chunking. However, more modules will also reduce the efficiency of the chunking.

## Acknowledgement

This work is partially supported by the ILMT project sponsored by TDIL program of MCIT, Govt. of India. We would like to thank all the members in Communication Empowerment Lab, IIT Kharagpur for their active participation in developing the resources required for this work.

## Reference

Bandyopadhyay, S. and Ekbal, A. (2006). *HMM Based POS Tagger and Rule-Based Chunker for Bengali.* In Proceedings of the Sixth International Conference on Advances In Pattern Recognition: pp. 384-390, Kolkata.

Bhat, R. A., and Sharma, D. M. (2011). *A Hybrid Approach to Kashmiri Shallow Parsing*. In LTC-2011: The 5th Language and Technology Conference: Human Language Technologies as a Challenge for Computer Science and Linguistics (LTC-2011).

Bharati, A., Chaitanya, V., and Sangal, R. (1995). *Natural Language Processing: A Paninian Perspective.* Prentice Hall India.

Dandapat, S. (2007) *Part Of Speech Tagging and Chunking with Maximum Entropy Model.* In Proceedings of IJCAI Workshop on "Shallow Parsing for South Asian Languages", Hyderabad, India. pp 29–32.

Das, D., Choudhury, M., Sarkar, S., and Basu, A. (2005). *An Affinity Based Greedy Approach towards Chunking for Indian Languages.* In Proceedings of ICON 2005 (NLP Association of India)

Das, D., and Choudhury, M. (2004). *Chunker and Shallow Parser for Free Word Order Languages: An Approach based on Valency Theory and Feature Structures.* Presented at the student paper competition of ICON 2004 (NLP Association of India).

Pal, S., and Bandyopadhyay, S. (2012). *Bootstrapping Method for Chunk Alignment in Phrase Based SMT.* In the Proceedings of the Joint workshop on exploiting Synergiesbetween Information Retieval and Machine Translation (ESIRMT) and Hybrid Approches to Machine Translation (HyTra), EACL-2012, pp.93-100, Avignon France.

Ray, P. R., Harish, V., Sarkar, S., and Basu A. (2003). *Part of Speech Tagging and Local Word Grouping Techniques for Natural Language Parsing in Hindi.* In Proceedings of International Conference on Natural Language Processing (ICON 2003). Mysore, India.

Vilain, M., and Day, D. (2000). *Phrase Parsing with Rule Sequence Processors: an Application to the Shared CoNLL Task.* In Proceedings of CoNLL-2000 and LLL-2000, Lisbon, Portugal.

De, S., Dhar, A., Biswas, S., and Garain, U. (2011). *On Development and Evaluation of a Chunker for Bangla.* In Proceedings of Second International Conference on Emerging Applications of Information Technology *(EAIT),* pp.321-324, 19-20 Feb.

Lafferty, J., McCallum, A., and Pereira, F. (2001). *Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data.* In *Proceedings of the Eighteenth International Conference on Machine Learning* (ICML-2001).