Role of Morphology Injection in SMT: A Case Study from Indian Language Perspective

SREELEKHA S, Indian Institute of Technology Bombay, India
PUSHPAK BHATTACHARYYA, Indian Institute of Technology Bombay, India

Phrase-based SMT is commonly used for automatic translation. However, PBSMT runs into difficulty when either or both of the source and target languages are morphologically rich. Factored models are found to be useful for such cases, as they consider word as a vector of factors. These factors can contain any information about the surface word and use it while translating. The objective of the current work is to handle morphological inflections in Hindi, Marathi and Malayalam using Factored translation models when translating from English. Statistical MT approaches face the problem of data sparsity when translating to a morphologically rich language. It is very unlikely for a parallel corpus to contain all morphological forms of words. We propose a solution to generate these unseen morphological forms and inject them into the original training corpus. We propose a simple and effective solution based on enriching the input with various morphological forms of words. We observe that morphology injection improves the quality of translation in terms of both adequacy and fluency. We verify this with experiments on three morphologically rich languages when translating from English. From the detailed evaluations we observed an order of magnitude improvement in translation quality.

Categories and Subject Descriptors: **• Computing methodologies → Artificial intelligence → Natural language processing → Machine translation • Computing methodologies → Artificial intelligence → Natural language processing → Phonology / morphology**

Additional Key Words and Phrases: Statistical Machine Translation, Factored Statistical Machine Translation Models, Morphology Injection.

1. INTRODUCTION

Formally, Machine Translation (MT) is a subfield of computational linguistics that investigates the use of software to translate text or speech from one natural language to another[1]. The MT methods are classified as transfer-based, rule-based, example-based, interlingua-based, statistics-based, etc. Statistical Machine Translation (SMT) is a MT paradigm where translations are generated on the basis of statistical models whose parameters are derived from the analysis of bilingual text corpora[2]. SMT approaches may include;

- **Word-based models**: The basic unit of translation is a word, where the ratio of the lengths of sequences of translated words tells how many foreign words each native word produces. IBM models 1 to 5 [Brown et. al., 1993] describe these models. Even though these models are simple, their biggest disadvantage is that they do not consider context while modeling.
- **Phrase-based models**: The aim is to reduce the restrictions of word-based models by translating chunks of words which are contiguous, also called phrases. Note that these phrases need not be linguistic phrases. The length of

This work is funded by Department of Science & Technology, Government of India under Woman Scientist Scheme (WOS A) with the project code – "SR/WOS-A/ET/1075/2014". Author's addresses: Sreelekha. S, DST-Woman Scientist, Dept. of Computer Science and Engineering, Indian Institute of Technology, Bombay, India, Email: sreelekha@cse.iitb.ac.in, Pushpak Bhattacharyya, Vijay & Sita Vashee Chair Professor, Dept. of Computer Science and Engineering, Indian Institute of Technology Bombay, India, Email: pb@cse.iitb.ac.in.

[1] http://www.mt-archive.info/
[2] http://www.statmt.org/book/

the phrase is variable. Phrase-based models are currently most widely used models for the SMT.

- **Syntax-based models**: Syntax-based translation is based on the idea of translating syntactic units, rather than single words or strings of words as in phrase based MT. These models make use of syntactic features of a sentence such as parse trees, parts of speech (POS) tags, etc.
- **Hierarchical phrase-based models**: Hierarchical phrase-based translation combines the strengths of phrase-based and syntax-based translation. It uses phrases (segments or blocks of words) as units for translation and synchronous context free grammars as rules (syntax-based translation).
- **Factored phrase-based models**: Factored models are a special case of phrase-based models. Factored models make use of vector of factors which may represent morphological or syntactic information about that phrase instead of just using surface form of phrase. Even though factored models try to add in linguistic support for statistical approach, data sparseness and increased decoding complexity are the big road blocks in their development.

Statistical translation models when translating into a morphologically rich language face two challenges:

- **Correct choice of inflection:** As single source root word can be mapped to several inflectional forms of target root word, the translation system should get the missing information from the source text that can help make correct inflectional choice.
- **Data sparsity:** During training, the corpus of the morphologically rich language does not have all inflectional forms of each word.

Most approaches to SMT, i.e., phrase-based models [Koehn, Och and Marcu, 2003], syntax-based models [Yamada and Knight 2001] do not allow incorporation of any linguistic information in the translation process. The introduction of factored models [Koehn and Hoang, 2007] provided this missing linguistic touch to the SMT. Factored models [Koehn and Hoang, 2007] treat each word in the corpus as vector of tokens. Each token can be any linguistic information about the word which leads to its inflection on the target side. Hence, factored models are preferred over phrase-based models [Koehn, Och and Marcu, 2003] when translating from morphologically poor language to morphologically richer language.

Factored models translate using *Translation* and *Generation* mapping steps. If a particular factor combination in these mapping steps has no evidence in the training corpus, then it leads to the problem of data sparseness. Though factored models give more accurate morphological translations, they may also generate more unknowns (words with respective translation is not present in the phrase table) compared to other unfactored models. In this paper, we study factored models and the problem of sparseness in the context of translation to morphologically rich languages.

For our experiments, we use three languages which carry different morphological levels: Hindi is morphologically complex compared to English with its post positions and pre positions; while Marathi is more complex with its suffix agglutination properties. On the other hand, Malayalam is highly complex with its word compounding and agglutination properties. However, Marathi and Hindi have some similarities except in Marathi there is agglutination of suffixes.

We consider an example of verb morphology in Hindi to understand the severity of the sparseness problem. Hindi verbs are inflected based on gender, number, person, tense, aspect and modality. Gender has two values (masculine, non masculine). Number has two values (singular and plural). Person has three values (first, second

and third). Tense has two values (present, non present). Aspect has three values (simple, progressive and perfect). Modality has around nine values (shall, will, can, etc.). Thus, for a single root verb in Hindi, we have 648 (2*2*3*2*3*9) inflected forms of it. Hence, a single English verb can be translated to 648 verb forms on the Hindi side. Hindi vocabulary has around 40,000 root verbs. Hence, in total 25,920,000 (648*40,000) verb forms. It is very likely that the parallel Hindi corpus cannot have all inflected forms of each verb. Also note that if the corpus size of the morphologically richer language is small, then the problem of sparseness will be more severe [Sreelekha et al. 2015].

Thus, even though we can use factored models to correctly generate morphological forms of words, the problem of data sparseness limits their performance. In this paper, we propose a simple and effective solution which is based on enriching the input corpora with various morphological forms of words. Application of the technique to an English-Hindi case study shows that the technique improves the translation quality and handles the problem of sparseness effectively.

The rest of the paper is organized as follows: Section 2 describes the related work; Section 3 describes the baseline factored translation model and the data sparseness problem; Section 4 discusses the Morphology Injection Technique; Section 5 discusses Morphology Generation process; Section 6 discusses Factor Generation process; Section 7 discusses Resource Generation process; Section 8 discusses Experiments and evaluations conducted; Section 9 gives a generalized solution to the sparseness problem; Section 10 draws conclusion and points to future work.

2. RELATED WORK

Since India is rich in linguistic divergence, there are many morphologically rich languages quite different from English as well as from each other. Hence, there is a large requirement for machine translation between them. Development of efficient machine translation systems using appropriate methodologies and with limited resources is a challenging task. There are many ongoing attempts to develop MT systems for Indian languages [Antony, 2013; Kunchukuttan et al., 2014; Sreelekha et al., 2014; Sreelekha et al., 2015; Sreelekha et al., 2016; Sreelekha et al., 2016] using both rule based and statistical approaches. In addition, there were many attempts to improve the quality of SMT systems such as using Monolingually Derived Paraphrases [Marton et al., 2009] or using Related Resource Rich languages [Nakov and Ng, 2012]. Considering the large amount of human effort and linguistic knowledge required for developing rule based systems, SMT systems became a better choice in terms of efficiency. Still the statistical systems fail to handle rich morphology.

There has been much work on translating from rich morphology languages into English compared to the work on translating from English into rich morphology languages [Habash and Sadat, 2006]. As per the studies, translating into morphologically richer languages is harder and more difficult than translating from morphologically richer languages [Koehn 2005]. For example, when translating from English, choosing the right inflected forms for the noun phrases based upon its role in the sentence makes the translation harder [Eleftherios and Koehn, 2008]. There have been various approaches to enrich the source in word based SMT such as; usage of part of speech tags by Uffing and Ney [2000], a post processing system with morphological and syntactical features by Minkov et al.[2007] etc. There has been

studies to enrich the target side when translating from English such as; usage of POS and morph stems (stemmed portion of words with morphological inflections) in the input along with morph tags on the target side proposed by Durgar El Kahlout and Oflazer [2006], case determination in Arabic proposed by Habash et al. [2007], Word Sense Disambiguation problem proposed by Carpuat and Wu [2007] etc.

Koehn and Hoang [2007] have conducted experiments on factored SMT models using morphology tags added on the morphologically rich side and scored with a 7 gram sequence model, along with POS tags for translating from English to German, Spanish and Czech. Birch et al. [2007] investigated the probabilistic models for using only source tags, where English was the target language. They have used Combinatorial Categorial Grammar (CCG) supertags as factors on the input words in factored SMT models. There were approaches by enriching the source language with grammatical features [Avramidis and Koehn, 2008] and appending the standard translation model with *synthetic phrases* [Chahuneau et al., 2013].

Although past work focuses on studying the complexity of factored translation models [Tamchyna and Bojar, 2013], the problem of data sparseness is not addressed, to the best of our knowledge. We discuss a case study in which we try to handle the noun/verb morphology in English to Hindi translation using factored models. There has been previous work done in order to solve the verb morphology for English to Hindi translation [Gandhe et al., 2011]. Gandhe et al. [2011] has used a classification approach based upon similarity and has augmented the phrase table with the verbal inflections. However they could get only an improvement of 1.5 BLEU score, since they have used an approach to generate the verb forms and then search for a matching in the phrase table which results in additional complexity. Our goal is to handle data sparseness against this case study. We have followed the nominal and verbal classification approach from Singh and Sharma [2010]. Our experiments show that the model performs very well in order to handle the noun and verb morphology for solving the sparseness problem.

3. BASELINE FACTORED TRANSLATION MODEL

3.1 General Factored model for handling morphology

Factored translation models allow additional annotation at the word level by considering word as a vector of tokens. Factored translation models can be seen as the combination of several components (language model, reordering model, translation steps, and generation steps). These components define one or more feature functions that are combined in a log linear model [Koehn and Hoang, 2007]:

$$p(e|f) = \frac{1}{Z}\sum_{i=1}^{n} \lambda_i h_i\,(e,f) \qquad (1)$$

From equation (1), each h_i is a feature function for a component of the translation, the λ_i values are weights for the feature functions and Z is the normalization constant.

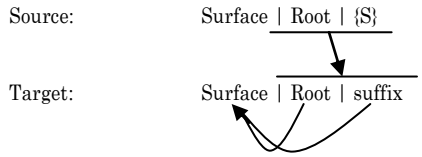


Figure 1: Factored model setup to handle inflections

Figure 1 shows a general factored model approach for translation from a morphologically poor language to a morphologically rich language. On the source side we have: Surface word, root word, and set of factors *S* that affect the inflection of the word on the target side. On the target side, we have: Surface word, root word, and suffix (can be any inflection). The model has the following mapping steps:

- **Translation step (T0):** Maps source root word and factors in *S* to target root word and target suffix
- **Generation step (G0):** Maps target root word and suffix to the target surface word. Note that the words which do not take inflections have *null* as values for the factors in *S*.

3.2. Problem of data sparsity

The SMT systems face the problem of data sparsity. One of the reasons is that data does not have enough inflectional forms for morphologically rich language when translating from a morphologically poor language to a morphologically rich language. Another reason is that data sparseness arises only when using factored models. We discuss these two reasons in detail in this section.

3.2.1. Sparsity when translating into a morphologically rich language

Root words in morphologically rich languages have many inflectional forms. When translating from a morphologically poor language to a morphologically rich language, a single word in the source language can be translated to multiple words in the target language. Unless training data has all such inflectional forms present, the model cannot generate correct inflectional form of the target word.

For example, assume the training data has the following pair of sentence:

boy plays → लड़का खेलता है (ladaka khelta hai)

Now, for any system trained with this data, for the test input as: "boy ate", the output would be: *लड़का खाया (ladaka khaya)*. This output is wrong, as it has incorrect inflection for the word "boy". Correct translation is: *लड़के ने खाया (ladake ne khaya).*

3.3. Sparsity while using Factored model

While factored models allow incorporation of linguistic annotations, this also leads to the problem of data sparsity. The sparsity is introduced in two ways:

- **Sparsity in Translation**: When a particular combination of factors does not exist in the source side training corpus

For example, let the factored model have single translation step: $X|Y \rightarrow P|Q$[3]. Suppose the training data has evidence for only $x_i|y_j \rightarrow p_k|q_l$ mapping. The factored model learnt from this data can not translate $x_u|y_v$, for all $u \neq i$ or $v \neq j$. The factored model generates "UNKNOWN" as output in these cases.

Note that, if we train a simple phrase based model only on the surface form of words, we will at least get some output, which may not have the correct inflectional markers, but it would still be able to convey the meaning.

- **Sparsity in Generation**: When a particular combination of factors does not exist in the target side training corpus

For example, let the factored model have single generation step: $P|Q \rightarrow R$.1 Suppose the target side training data has an evidence of only $p_a|q_b \rightarrow r_c$. The factored model learnt from this data can not generate from $p_u|q_v$ all $u \neq a$ or $v \neq b$. Again the factored model generates "UNKNOWN" as output.

[3] Capital letters indicate factors and small letters indicate values that corresponding factors can take

Thus, due to sparsity, we cannot make the best use of factored models. In fact, they fare worse than the phrase based models, especially, when a particular factor combination is absent in the training data. In the case of noun inflection factored model, this can be observed through following example:

Consider following sentence to be the training data.

- *Factored: boys|boy|plural|direct play|play|.|. → लड़के|लड़का|-e खेलते|.|. हैं|.|. (ladake khelte hain)*
- *Unfactored: boys play → लड़के खेलते हैं (ladake khelte hain)*

Now, let the test input be: *boys|boy|plural|oblique (for factored) or boys (for unfactored).* As factor combination *boy|plural|oblique* is absent in the training data of the factored model, it will generate unknown output. Whereas, even though morphologically wrong phrase based model will generate *लड़के (ladke) (boys)* as output. Thus the use of factored models may lead to low quality translation.

3.4 Basic Morphology Injection for solving data sparsity

The reason for data sparseness in factored models is either the combination of source side factors or target side factors that are not present in the training data. So, is it possible to get all the combinations of factors in the training data? In our case, we are using three factors on source side, i.e., lemma, number and direct/oblique case and one factor on the target side, i.e., root word (Note that, root word here is used for a noun with no morphological inflection, e.*g., लड़का (ladka) (boy)).* There is no generation step in our mapping; hence, sparseness due to generation step is already avoided. To handle the sparseness due to translation step, we need to have all morphological forms of each root word in the training data.

Section 4 explains the morphological classification of nouns based on number, direct/oblique case and class of the noun. Classification table in Figure 2 shows the suffix that a particular noun takes based on these three factors. We need to have number, case and class of the noun to be known on English side (as we are translating from English) to generate all morphological forms of a given root word in Hindi. In the Section 5, we describe how to morphologically classify nouns and to generate number and case factors for nouns. Gandhe et al., [2011] try to handle verbal inflections using similar technique in which they classify verbs into different classes. Each class has verbs which take similar inflections. After classification, they generate all the inflectional forms of verbs depending upon the class of the verb.

4. MORPHOLOGY INJECTION TECHNIQUE

As discussed in the Section 3, we need to generate all combinations of the factors used to handle the sparseness of factored models. In this section we present a morphology injection method that generates various morphological forms of noun entities and augments the training data with newly generated morphological forms of nouns as additional parallel sentences.

The basic algorithm of the morphology injection method can be described as below:

1. Find out the noun entity pairs (English-Hindi)
2. Classify Hindi nouns into classes
3. Generate new morphological forms of the nouns using the classification table
4. Augment the training data with the generated new forms.

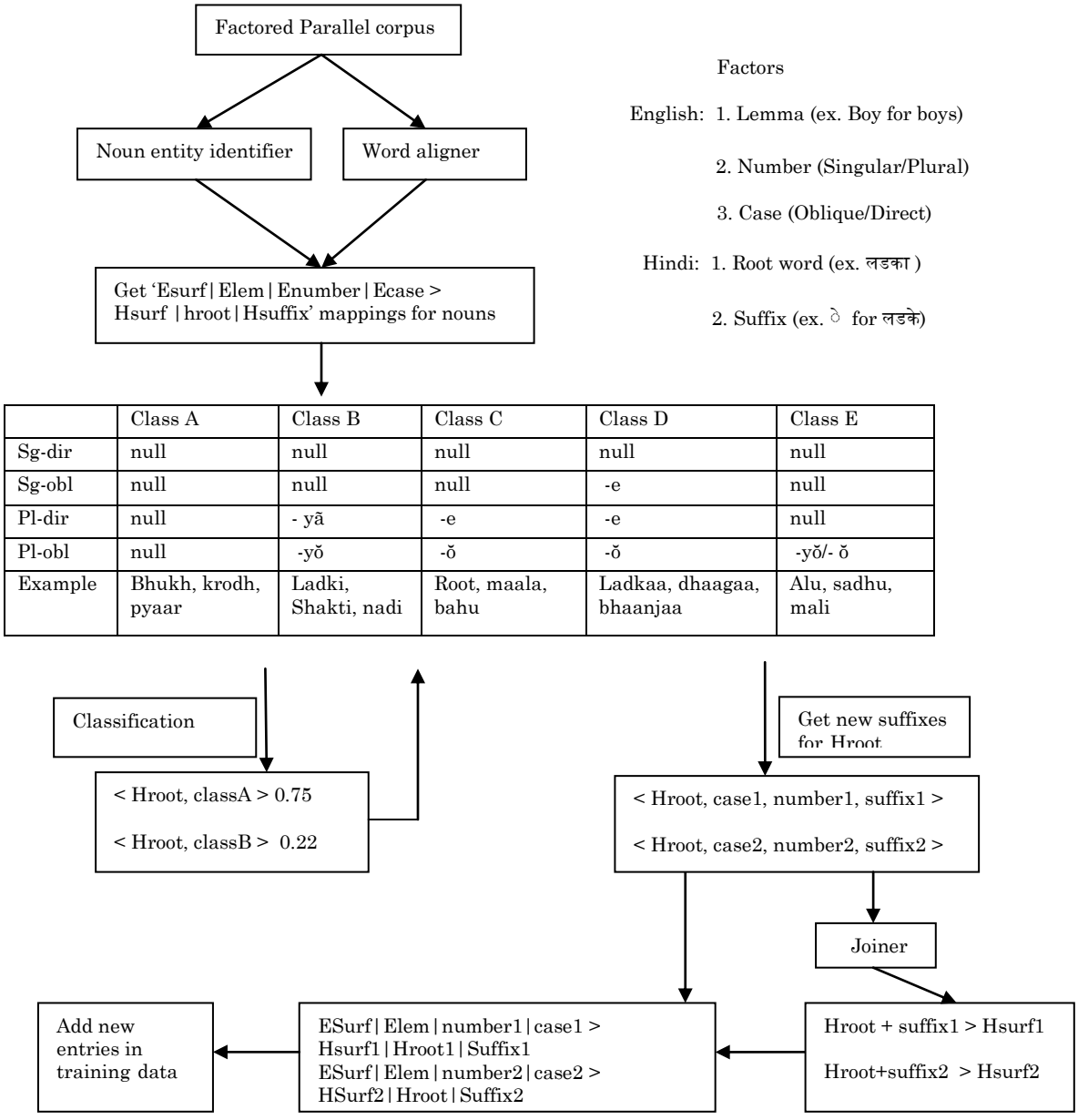
For example, let the noun pair be *'river - नदी (nadi)'*. Class of Hindi noun *नदी (nadi)*

is B. Now, we generate new forms of *नदी (nadi)* using the classification table shown in Figure 2.

river | sg | dir – नदी (nadi) | नदी (nadi) | Null
river | sg | obl – नदी (nadi) | नदी (nadi) | Null
river | pl | dir नदियाँ (nadiya) | नदी (nadi) | याँ (yam)
river | pl | obl – नदियों (nadiyom) | नदी (nadi) | यों (yom)

The algorithm is elaborated in the following subsections, where it is used in two different contexts.

4.1. Using parallel factored corpus



	Class A	Class B	Class C	Class D	Class E
Sg-dir	null	null	null	null	null
Sg-obl	null	null	null	-e	null
Pl-dir	null	- yă	-e	-e	null
Pl-obl	null	-yŏ	-ŏ	-ŏ	-yŏ/- ŏ
Example	Bhukh, krodh, pyaar	Ladki, Shakti, nadi	Root, maala, bahu	Ladkaa, dhaagaa, bhaanjaa	Alu, sadhu, mali

Figure 2: Using parallel factored corpus for Morphology injection method

We can use a parallel factored corpus which has lemma, number and direct/oblique case factors on English side and root word factor on Hindi side. Factors are generated as described in Section 6. We need to have factored English-Hindi corpus with factors as shown in Figure 2. We pass the corpus to noun entity identifier, which is based on the POS tags to get the noun entities present in the corpus. We align the corpus word by word to find out the pairs of nouns in English-Hindi corpus. So, now we get the mappings of the form: Esurf|Elem|Enumber|Ecase → Hsurf|Hroot|Hsuffix[4] for each noun pair. We classify these noun pairs using Enumber, Ecase and Hsuffix as will be discussing in Section 5**.**

As each noun pair will have many corresponding combinations of number, case and suffix in the training data, we need to predict the probability of the noun being classified into each of the five classes. This can be simply done by keeping a count for each class for a given noun pair and classifying each occurrence of this pair in training data into one of the classes. Note that there may be cases when the noun pair cannot be classified or there can be multiple classes into which the pair can be classified. Then we need to increase the count of each class. Then, the counts can be normalized to get the probability. Finally, the noun pair can be classified into a class which has the highest probability.

We can get all the combinations of number, case and suffix for a given noun pair from the classification table after we classify the noun pair. We use these new suffixes to generate new inflected forms of the root word in Hindi. We pass new suffixes and the Hindi root word to the Joiner tool, which generates new surface forms. For example, given *लड़का (ladka) (boy)* and *'-e'* Joiner will generate *लड़के (ladke) (boys)*. Details of the Joiner tool are discussed in Section 7.2. Finally, we get new factored pairs of the form: *Esurf|Elem|Enumber'*[5]*|Ecase' → Hsurf'|Hroot|Hsuffix'*. These new pairs are added to the original training data.

4.2 Using monolingual lexicon

We use the Hindi lexicon in our case. The Hindi lexicon contains Hindi nouns, proper nouns, adjectives, verbs, etc. Figure 3 shows the pipeline of the same. The pipeline is somewhat similar to that in Figure 2, but here, instead of predicting the class of the noun pair from its suffix, we actually classify the Hindi noun into one of the five classes. We need the gender information to classify a Hindi noun into a morphological class, whether or not it takes inflections and its ending characters. We classify the nouns present in the lexicon as shown in Figure 3 using this information. We generate morphologically inflected forms of the Hindi noun using the classification table shown in Figure 2. We also generate the English counterpart of the Hindi noun. We use the Hindi to English dictionary for the same. After getting English side root word, we generate pairs of the form: .|*Elem*|*Enumber'*|*Ecase'* → *Hsurf'*|*Hroot*. Since, we cannot generate English surface word form; it is denoted by a dot in the mapping. This does not affect the factored model settings, as our translation step does not use English surface word. We then append the original training data with the newly generated pairs. Note that the factored settings mentioned above is different from that the one described in Section 5, as we do not use the Hindi side suffix here.

[4] Surface form, lemma, number and case factors for English noun; Surface form, root form and suffix for respective Hindi noun

[5] The apostrophes after factors indicate that they are the newly generated factors.

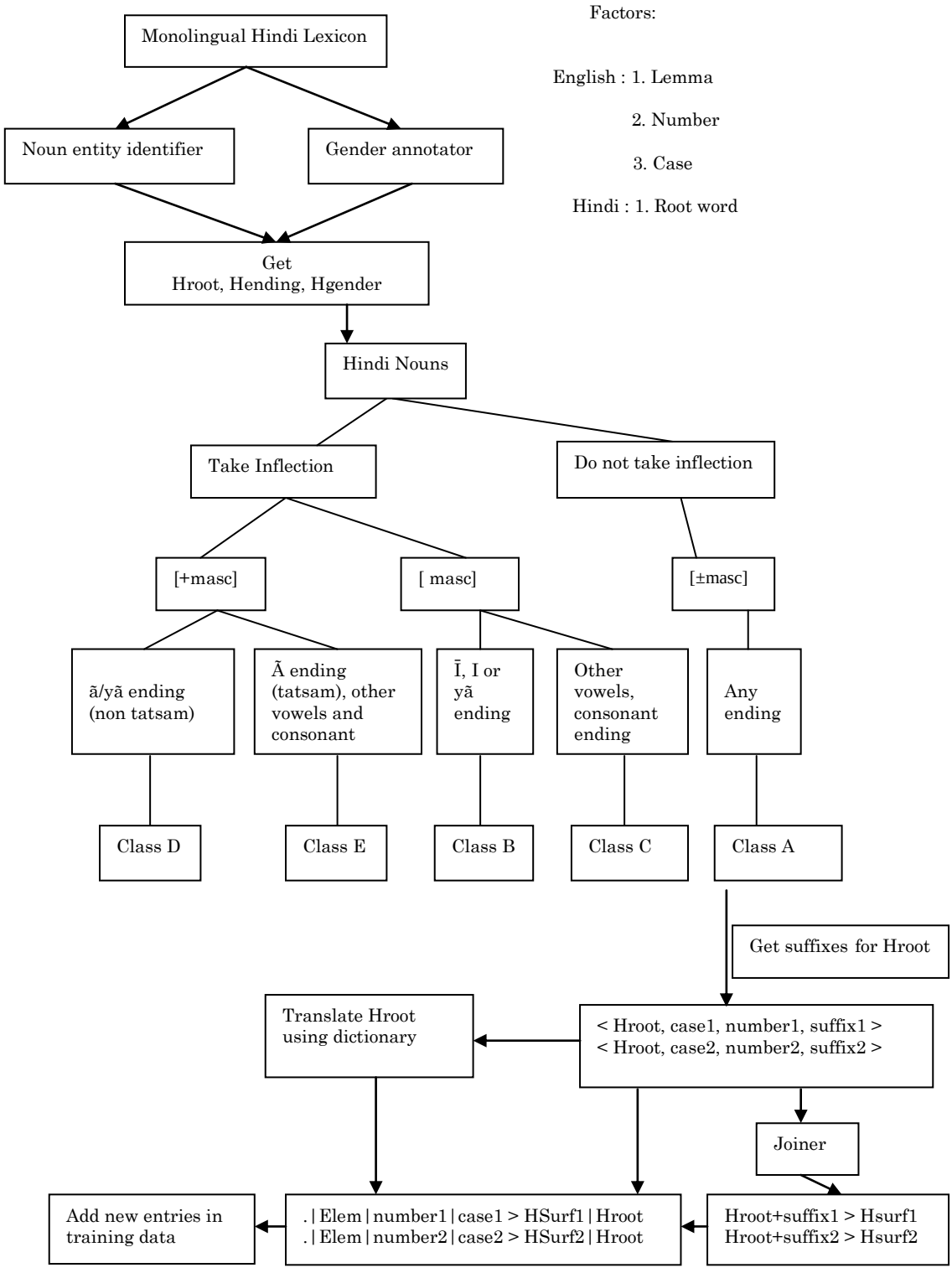


Figure 3: Using monolingual Hindi lexicon for Morphology injection method

5. MORPHOLOGY GENERATION

Hindi is a morphologically richer language compared to English. It has morphological inflections on nouns and verbs. In addition, adjectives in Hindi show the inflection according to the gender and number of the noun it modifies. In this section, we describe the problem of handling noun and verb morphology when translating from English to Hindi using factored models. We do not consider the inflections on adjectives in this work, since adjectives take inflections only in some cases, where it ends with *आ (aa);* such as the adjective *अच्छा (achcha)* can take forms like *अच्छे (achche), अच्छी (achchi)* etc.

5.1 Noun morphology

In this section, we discuss the factored model for handling Hindi noun morphology and the solution to the data sparseness problem. Hindi nouns show morphological inflections only for number and case. The number can be either singular or plural and the case marking on Hindi nouns is either direct or oblique. Gender, an inherent lexical property of Hindi nouns (masculine or feminine) is not morphologically marked, but is realized via agreement with adjectives and verbs. Morphological classification of the noun into five classes is shown in Figure 2. All nouns in the same class have the same morphological inflection behavior [Singh and Sarma 2010].

5.1.1 Predicting Inflectional Class for New Lexemes

We need gender information for the classification of new lexemes into one of the five classes as shown in Figure 3. Its inflectional class can be predicted using the procedure outlined in Singh and Sarma [2010] after gender is lexically assigned to the new lexeme. A masculine noun may or may not be inflected based on its semantic property. If it is an abstract noun or a mass noun it will fall into the non inflecting Class A irrespective of its phonological form. On the other hand, a countable lexeme will fall into one of the two masculine classes based on its phonological form. A similar procedure follows for feminine nouns.

5.1.2 Factored model setup to handle noun morphology

If we decide to use factored models for handling noun inflections, it is very natural to use number and case as factors on the English side. The factored model mapping to handle noun inflections is shown in Figure 4. The generation of the number and case factors is discussed in Section 6. So, in this case, the set S, consists of number and case.

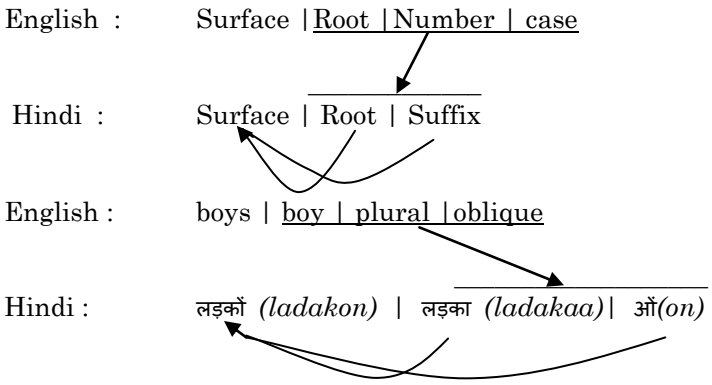


Figure 4: Factored model setup to handle nominal inflections

5.1.3 Building word form dictionary

In the case of factored model described in Section 5.1.2: the factored model has a single translation step and single generation step:

- **Translation step:** Map root noun, number, and direct/oblique case on English side to root noun and suffix on Hindi side. To solve the sparseness in the translation step, we need to have all English pairs of the form *root*|*number*|*case*→*Hindi noun root*|*number*|*suffix* present in the training data.
- **Generation step:** Map root noun and suffix to surface form on Hindi side. To solve the sparseness in generation step, we need to have all *Hindi noun* pairs of the form *root*|*number*|*suffix* → *Hindi surface word* present in the training data.

In other words, we need to get a set of suffixes and their corresponding number case values, for each noun pair. We need to generate Hindi surface words to remove sparseness in the generation step using these suffixes and the Hindi root word. Also, we need to generate four pairs for each noun present in the training data, i.e., *(sg-dir, sg-obl, pl-dir, pl-obl*) and get their corresponding Hindi inflections. In the following section, we discuss how to generate these morphological forms.

5.1.3.1. Generating new morphological forms:

Figure 2 from Section 4 shows the pipeline to generate new morphological forms for an English-Hindi noun pair. We need to know the suffix of a noun in Hindi for the corresponding number and case combination to generate the different morphological forms. We use the classification table shown in Figure 4 for the same. Nouns are classified into five different classes, namely A, B, C, D, and E according to their inflectional behavior with respect to case and number [Singh et al., 2010]. All nouns in the same class show the same inflectional behavior. To predict the class of a Hindi noun, we develop a classifier which uses gender and the ending characters of the nouns as features [Singh et al., 2010]. We get four different suffixes and corresponding number-case combinations using the class of Hindi noun and classification shown in Figure 2. For example, if we know that the noun *लड़का (ladkaa) (boy)* belongs to class D, then we can get four different suffixes for *लड़का (ladkaa) (boy)* as shown in Table 1.

Table 1: Morphological suffixes for boy- लड़का *(ladakaa)* noun pair

English root|**Number**|**Case**	**Hindi root**|**Suffix**
boy|singular|direct	लड़का *(ladakaa)*|null
boy|singular|oblique	लड़का *(ladakaa)*| ए *(e)*
boy|plural|direct	लड़का *(ladakaa)*| ए *(e)*
boy|plural|oblique	लड़का *(ladakaa)*| ओं *(a)*

5.1.3.2. Generating surface word:

Next we generate the Hindi surface word from the Hindi noun root and the suffix using a rule based joiner (reverse morphological) tool as described in Section 7. The rules of the joiner use the ending of the noun root and the class to which the suffix belongs as features. Thus, we get four different morphological forms of the noun

entities present in the training data. We augment the original training data with these newly generated morphological forms. Table 2 shows four morphological forms of "*boy- लड़का (ladakaa)*" noun pair. Note that the joiner solves the sparseness in generation step.

Table 2: New morphological forms of boy लड़का (ladakaa) noun pair

English root|Number|Case	Hindi surface|Root|Suffix
boy | singular | direct	लड़का *(ladakaa)* | लड़का *(ladakaa)* | null
boy | singular | oblique	लड़के *(ladake)* | लड़का *(ladakaa)* | ए (e)
boy | plural | direct	लड़के *(ladake)* | लड़का *(ladakaa)* | ए (e)
boy | plural | oblique	लड़कों *(ladakon)* | लड़का *(ladakaa)* | ओं *(on)*

5.2 Verb morphology

In this section, we discuss the factored model for handling Hindi verb morphology and the data sparseness solution in the context of the same. Many grammarians and morphologists have discussed the inflectional categories of Hindi verbs but these studies are either pedagogical or structural in approach. Across all approaches, there is much agreement on the kinds of inflectional categories that are seen in Hindi verbs. The inflection in Hindi verbs may appear as suffixes or as auxiliaries. These categories and their exponents are described in Singh and Sarma[2011]. When translating from English to Hindi, to handle these verbal inflections, we need all the factors available with us to implement a factored model.

5.2.1 Factored model setup to handle verb morphology

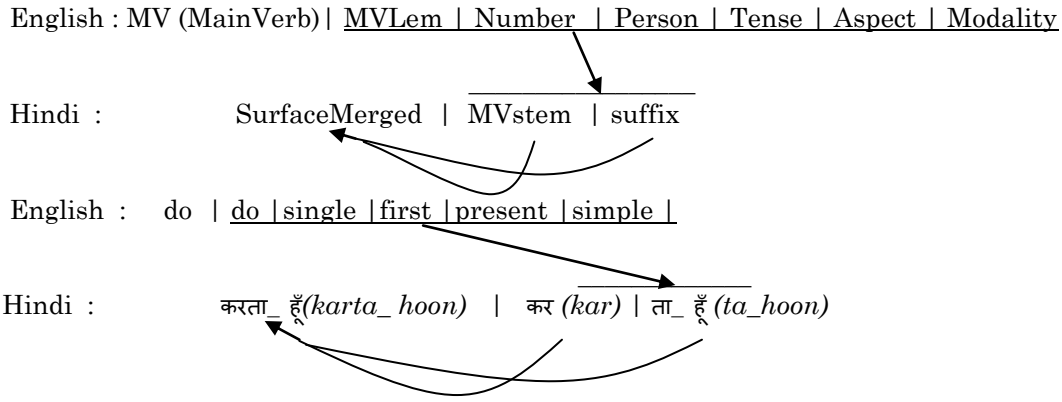


Figure 5: Factored model mapping for handling verbal inflections in Hindi

Verbal inflections in Hindi depend on tense, number, person, gender, aspect and modality [Singh and Sarma, 2011]. English verbs do not explicitly carry this information. Hence, when translating from English to Hindi, we need to consider syntactic and semantic information hidden in the English sentence to get this information, apart from the original verb. Once we get these factors we can use the factored model mapping shown in the Figure 5 to handle the morphological inflections of Hindi verbs. Gender is not used in the mapping due to two reasons.

Firstly, getting gender information on English side is very hard. Secondly, just using many factors in factored model does not improve the results, but instead it may result in degradation. Hence, we tried using some of these factors which are important and which are easily available. On the English side, we only use the lemma of main verb and remove any auxiliary verbs present. Information contained in the auxiliaries and inflection of the verb will already be present in the other factors that we are using in factored model. For example, if a sentence has the verb '*is doing'*, we remove '*is'* and retain the lemma of the word '*doing*', i.e., '*do'*.

Hence, set S, consists of number, person, tense, aspect and modality. Example of factors and mapping steps are shown in Figure 5. The generation of the factors is discussed in Section 6. Here in this case, the verb *"doing"* will be having different inflectional forms in first person, second person and third person of the subject. On the Hindi side, we create a merged verb form from the main verb and auxiliary verbs. The main verb stem is used as a factor. We merge inflections from the main verb with auxiliaries and create another factor.

5.2.2 Building word form dictionary

Thus, in the case of factored model described in Section 5.2.1: the factored model has a single translation steps and single generation step:

- **Translation step:** Map main verb lemma, number, person, tense, aspect, and modality on English side to main verb stem and merged suffix on Hindi side. To solve the sparseness in the translation step, we need to have all English verb pairs of the form *root*|*numer*|*person*|*tense*|*aspect*|*modality* →*Hindi verb root*|*suffix* present in the training data.
- **Generation step:** Map main verb stem and merged suffix to surface form on Hindi side. To solve the sparseness in the generation step, we need to have all *Hindi verb* pairs of the form *root*|*suffix* → *Hindi surface word* present in the training data.

In other words, we need to get a set of suffixes and their corresponding number person tense aspect modality values, for each verb pair. Using these suffixes and the Hindi root word, we need to generate the Hindi surface words to remove sparseness in the generation step. In the Section 5.2.2.1, we discuss how to generate these morphological forms.

5.2.2.1 Generating new morphological forms:

Table 3: Suffixes for Hindi verbs based on number, person, tense and aspect

		Simple	
		Singular	Plural
Present	First	ता हूँ / ती हूँ *(ta hoon / ti hoon)*	ते हैं / ती हैं *(te haen) / ti haen)*
	Second	ता है / ती है *(ta hae / ti hae)*	ते हो / ती हो *(te ho / ti ho)*
	Third	ता है / ती है *(ta hae) / (ti hae)*	ते हैं / ती हैं *(te haen) / (ti haen)*

Table 3 shows a subpart of a table which is used to gets suffixes for Hindi verb roots.

Note that no pre-classification of verbs is required, as these suffixes apply to all verbs. Table 4 shows few of many suffixes for *भाग (bhaag)*.

Table 4: Morphological suffixes for run-भाग *(bhaag)* verb pair based on number (N), person (P), tense (T), aspect (A) and modality (M)

English root |N|P|T|A|M	**Hindi root | Suffix**
run |singular |first |present |simple |	**भाग | ता हूँ / ती हूँ** *(bhaag |ta hoon / ti hoon)*
run |plural |first |present |simple |	**भाग | ते हैं / ती हैं** *(bhaag | ta haen / ti haen)*
run |singular |second |present |simple |	**भाग | ता है / ती है** *(bhaag |ta hae / ti hae)*
run |plural |second |present |simple |	**भाग | ते हो / ती हो** *(bhaag | te ho / ti ho)*
run |singular |third |present |simple |	**भाग | ता है / ती है** *(bhaag | ta hae / ti hae)*
run |plural |third |present |simple |	**भाग | ते हैं / ती हैं** *(bhaag | te haen / ti haen)*

5.2.2.2 Generating surface word:

For verbs, we generate the Hindi surface word from the Hindi verb root and the suffix using the rule based joiner tool as described in Section 5.1.3.2. Here, the joiner uses the verb root's end as the features to generate different morphological forms of the verb entities. Table 5 shows morphological forms of *run-भाग (bhaag)* verb pair.

Table 5: New morphological forms of run भाग (bhaag) verb pair

English root |N|P|T|A|M	**Hindi surface | Root | Suffix**
run |singular |first |present |simple |	**भागता हूँ / ती हूँ | भाग | ता हूँ / ती हूँ** *(bhaagta hoon / ti hoon | bhaag |ta hoon / ti hoon)*
run |plural |first |present |simple |	**भागते हैं | भाग | ते हैं /** *(bhaagte haen | bhaag | te haen)*
run |singular |second |present |simple |	**भागता है / ती है | भाग | ता है / ती है** *(bhaagta hae / ti hae | bhaag |ta hae / ti hae)*
run |plural |second |present |simple |	**भागते हो / ती हो | भाग | ते हो / ती हो** *(bhaagte ho / ti ho | bhaag | te ho / ti ho)*
run |singular |third |present |simple |	**भागता है | भाग | ता है / ती है** *(bhaagta hae | bhaag | ta hae / ti hae)*
run |plural |third |present |simple |	**भागते हैं | भाग | ते हैं / ती हैं** *(bhaagte haen | bhaag |te haen / ti haen)*

5.3 Noun and Verb morphology

Finally, we create a new factored model which combines factors on both nouns and verbs, as shown in Figure 4 and 5. We build word form dictionaries separately as discussed in Section 5.2 and Section 5.3. Then, we augment training data with both the dictionaries. Note that, factor normalization[6] on each word is required before this step to maintain the same number of factors. We also created a word form dictionary for the phrase based model. We follow the same procedure as described in Section 5, but we have removed all the factors from the source and target words except the surface form.

6. FACTOR GENERATION

In this Section we discuss the generation of factors with the help of syntactic and morphological tools. We extract the number and case of the English nouns and number, person, tense, aspect, modality of the English verbs as follows:

Noun factors:

- **Number factor:** We use *Stanford POS tagger*[7] to identify the English noun entities [Toutanova et al., 2003]. The POS tagger itself differentiates between singular and plural nouns by using different tags.
- **Case factor:** It is difficult to find the direct/oblique case of the nouns as English nouns do not contain this information. Hence, to get the case information, we need to find out features of an English sentence that correspond to direct/oblique case of the parallel nouns in Hindi sentence. We use object of preposition, subject, direct object, tense as our features. These features are extracted using semantic relations provided by Stanford's typed dependencies [De Marneffe et al., 2008].

Verb factors:

- **Number factor:** Using typed dependencies we extract the subject of the sentence and get number of the subject as we get it for a noun.
- **Person factor:** We do lookup into simple list of pronouns to find the person of the subject.
- **Tense, Aspect and Modality factor:** We use POS tag of verbs to extract tense, aspect and modality of the sentence.

6. 1 Using semantic relations to generate the factors

We need to generate tense, person, number and gender information of the verb on English side. Since this information is absent in the raw sentence, we need deep information about the sentence, such as POS tagging, semantic relations, parse tree, etc. to generate this information. In the following subsections, we will explain how to make use of these extra resources to get tense, person, number and gender information. We use the Stanford dependency parser for getting the syntactic parse tree of the sentence. We also use the semantic relations provided by Stanford's typed dependencies [Marneffe et al. 2008]. The current representation contains approximately 53 grammatical relations as described in Marneffe et al. [2008]. The

[6] Use *null* when particular word cannot have that factor
[7] http://nlp.stanford.edu/software/tagger.shtml

dependencies are all binary relations: a grammatical relation that holds between a governor (also known as a regent or a head) and a dependent.

6.1.1. Tense factor

Algorithm 1 describes how to get tense, aspect and modality of the sentence using a parse tree. The expansion of tags used are shown below,

POS – Possessive ending
VB – Verb, base form
VBD – Verb, past tense
VBZ – Verb, 3rd person singular present
MD – Modal
VBN – Verb, past participle
NN – Noun, singular or mass

Algorithm 1 Get tense, aspect and modality of the sentence

Input: Parse tree of the sentence
Output: Tense, aspect and modality of the sentence

```
tense, aspect, modality =Empty array of strings
For each leaf node in parse tree:
    POS = parent (leaf) Ψ //parts of speech (POS) tag of leaf
    if   (POS == "VBP"   || POS == "VBZ"   || POS == "VB")
            add "present" to tense
    else if (POS == "VBD")
            add "past" to tense
    else if (POS == "MD")
            if ( leaf == "could" && leaf == "would")
                    add "past" to tense
            else if ( leaf == "will" && leaf == "shall")
                    add "future" to tense
            else
                    add "present" to tense
    else if (POS == "VBG")
            add "progressive" to aspect
    else if (POS == "VBN")
            add "perfect" to aspect
return tense, aspect, modality
```

Algorithm 2 uses typed dependency to get the subject of the sentence. Person of the subject is found by comparing subject with pronouns. If the subject is not a pronoun, then most probably it will be in third person.

6.1.2 Person Factor

Algorithm 2 Get person of the sentence

Input: Parse tree of the sentence, Typed dependencies
Output: Person of the sentence

```
1: person=Empty string
2: subject = get subject using typed dependency "nsubj"
3: if (subject in ["i", "we"])
4:         person= "first"
5: else if (subject in ["you"])
6:         person= "second"
7: else if (subject in ["he", "she", "it", "they"])
8:         person= "third"
9: else
10:        person= "third"
11: return person
```

Algorithm 3 describes how to use parts of speech (pos) tag of subject to get the number of subject. If POS tag end with s, then subject is plural, otherwise it is singular.

6.1.3 Number Factor

Algorithm 3 Get number of the subject in the sentence

Input: Parse tree of the sentence, Typed dependencies
Output: Number of the subject

```
number =Empty string
subject = get subject using typed dependency "nsubj"
POS = parent(subject)Ψ//parts of speech (POS) tag of subject
 if (POS.startsWith("NN") && POS.endsWith("S"))
        number= "plural"
else if (POS.startsWith("NN") && ! POS.endsWith("S"))
        number= "singular"
return number
```

Algorithm 4 describes how to get the gender of the subject of the sentence. However, this algorithm is very weak since it gets gender by comparing subject with few pronouns. Hence, other pronouns and most importantly proper nouns are not classified.

6.1.4 Gender Factor

Algorithm 4 Get gender of subject of the sentence

Input: Parse tree of the sentence, Typed dependencies
Output: Gender of the sentence

```
1:  gender =Empty string
2: subject = get subject using typed dependency "nsubj"
3: if (subject in ["he"])
4:         person= "+musc"
5: else if (subject in ["she"])
6:         person= " musc"
7: else if (subject in ["it"])
8:         gender= "neutral"
9:  return gender
```

To get the direct/oblique case of nouns on English side, we need to find out features of an English sentence that correspond to the direct/oblique case of nouns in Hindi. Currently, we use following two features for this purpose.

- Object of preposition has Oblique case

For example, Fishes live in the rivers
मछलियाँ नदियों में रहती हैं
{machhaliyan nadiyon me rahti hain}
{ fishes rivers in live}

Here, *नदियों (nadiyon)* is oblique form of *नदी (nadi)*. In the English sentence, *river* is an object of the preposition *in*. Hence, we can say that the object of preposition in English sentence corresponds to an oblique case of that object in parallel Hindi sentence.

- Subject of the sentence is oblique if it has a direct object and tense of the sentence is past, present perfect or past perfect

For example, *Boys ate apples*
लड़कों ने सेब खाए
{ladkon ne seb khaye}
boys apples ate

Here, *लड़कों* is oblique form of *लड़का*. In the English sentence, '*boys'* is the subject of the sentence. It has a direct object, *apples*. Also, sentence has past tense.
Consider another example:

Boys went to school
लड़के पाठशाला गए
{ladke pathshala gaye}
boys school went

Here, *लड़के (ladke)* is the direct form of *लड़का (ladka)* as it is plural. (Note that, direct form of *लड़का (ladka)* when plural and oblique form of *लड़का (ladka)* when singular, are same, *i.e., लड़के (ladke*). In the English sentence boys is the subject of the sentence. But it does not have a direct object.

Algorithm 5, implements above two features to get the case of nouns by using tense from Algorithm 1. In order to get a detailed analysis we have retrieved information of tense, aspect, modality separately in Algorithm 1. However for computational purpose, we have combined the verb's tense and aspect together in a variable and that variable has been used as tense factor in Algorithm 5.

6.1.4 Case Factor

Algorithm 5. Get direct/oblique case of the nouns in the sentence

Input: Parse tree of the sentence, Typed dependencies, subject, direct Object, tense
Output: Case of the nouns

```
1: case=Empty Map of strings
2: if (subject != " " && directObject != " ")
3:      if (tense == "past" || tense == "past perfect" || tense == "present perfect")
4:           Put (subject, "oblique") in case
5: For each entry dep in typed dependencies:
6:      //Object of preposition has "oblique" case
7:      if  (dep.startsWith("prep")  ||  dep.startsWith("prepc"))
8:           Put (getObject(dep), "oblique") in case
```

9: For all other nouns in the sentence:
10: Put (noun, "direct") in case
11: return case

7. RESOURCE GENERATION

In this Section we discuss the resources that need to be built before actual training of the translation system starts.

7.1 Classification technique used

The approach of using parallel factored corpus as discussed in Section 4.1 is error prone and also it depends on the accuracy of the classification technique. We had Hindi lexicon readily available with us. Hence, we went forward with the approach of using Hindi lexicon for Morphology injection for the English-Hindi pair. The available Hindi lexicon size is 113,266 words. The lexicon has words classified into their morphological classes. Hence, we easily generated new combinations of factors, i.e., case, number and suffix for Hindi nouns as we have discussed in Section 5.

We have used a combined parallel factored corpus approach for generating forms in both Malayalam and Marathi using the various paradigm classification rules. We have created morphological rules and generated the inflectional forms from a parallel root word list which is extracted from a parallel corpus as well as from lexicon.

7.2 Development of a Joiner tool

After getting new suffixes for the Hindi root word, we need to form surface word by joining root word and suffix. We developed a rule based joiner (or reverse morphological) tool, which merges the root and the suffix based on the class to which the suffix belongs and the root word ending. Some of the rules are described below:

```
if (suffix in [यों, याँ] (yom, yam))
        if (ending in [ -e, इ, ई, ँ, ं])
                return ( root + suffix)
        else if (ending in [-ee])
                return (root - ending +  -e + suffix)
```

For example, if the input to joiner is: *नदी (nadi) (river)* and *यों (yom) (s*), then above rule matches for the given input. As *नदी (nadi) (river)* ends in -ee, output will be root ending + -e + suffix, i.e., *नदियों (nadiyon) (rivers)*. Similar rules are formed for other suffixes and classes.

7. 3 Development of a dictionary

After getting new morphological forms for the Hindi root forms of the nouns, we were in need of a dictionary to translate these nouns from Hindi to English. We already had a dictionary which contained 128,241 Hindi English pairs of words. But, the noun entities present in both the Hindi lexicon and the dictionary were only 9,684. Hence, instead of using this dictionary, we decided to go with an alternative approach, where we use Google's freely available online translation system[8] to

[8] https://translate.google.com/

generate English nouns from Hindi. While doing this, we encountered a problem of infrequent nouns in Hindi. There were many Hindi nouns in the lexicon that were translated into same English noun. *E.g. लड़का (ladka) (boy) and छोरा (chhora) (boy)* are translated to boy. *मछली (machhali) (fish) and मच्छी (machchhi) (fish)* are translated into fish. If we use these pairs as it is, there is a possibility of degrading translation as English noun may get translated to an infrequent word.

To solve the problem of infrequent words, we simply do two passes of the translation. In first pass, we translate nouns in Hindi lexicon using translation system. In the second pass, we translated these translations back to Hindi using the same translation system. Hence, we get new Hindi lexicon in which the infrequent nouns are eliminated. We use these new pairs as a dictionary to translate the Hindi root words. Note that if one has frequencies of the nouns in the lexicon, they can be used directly to eliminate infrequent nouns.

7. 4 Development of a Factor generation tool

We developed a factor generation tool to generate tense, person, number and gender information of the verb on English side using the Stanford parser and typed dependencies in the same way as described in Section 6.

8. EXPERIMENTS AND EVALUATION

We performed the experiments on ILCI (Indian Languages Corpora Initiative) English-Hindi (En- Hi), English-Marathi (En-Mr) and English-Malayalam (En-Ml) corpus. The domain of the corpus is health and tourism. We used 46,000 sentence pairs for training and 3000 sentence pairs for testing. Table 6 shows the statistics of the corpus used for training, testing and tuning. The word form dictionary was created using the Hindi word lexicon. Table 7 shows the statistics of the generated word form dictionary. We used *GIZA++* for finding out the word-alignments and *Moses* toolkit [Koehn et al., 2007] was used for training and decoding. The Language model was trained on the target corpus with IRSTLM [Federico et al., 2007].

Table 6: Statistics of the corpus used

Sl. No	Corpus Source	Corpus Domain	Training Corpus Size [Parallel Sentences]	Tuning (MERT) Corpus Size [Parallel Sentences]	Testing Corpus Size [Parallel Sentences]
1	ILCI	Health	23000	500	1500
2	ILCI	Tourism	23000	500	1500
Total			46000	1000	3000

Table 7: Statistics of the generated word form dictionary

Language	Verb forms generated	Noun forms generated	Total word form dictionary size
Hindi	390392	202544	592936
Marathi	106570	54762	161332
Malayalam	280000	125672	405672

Our baseline system is a simple factor based model as described in Section 3. We have used factored model setup for noun and verb morphology respectively as

described in Section 5 for experiments. We compared the translation output of the following systems:

- Phrase-based (unfactored) model (Phrase)
- Basic factored model for solving noun and verb morphology (Fact)
- Phrase-based model trained on the corpus used for Phrase augmented with the generated word form dictionary for solving noun and verb morphology (Phrase-Morph)
- Factored model trained on the corpus used for Fact augmented with the generated word form dictionary for solving noun and verb morphology (Fact-Morph)
- We have also conducted experiments by injecting noun and verb factors separately in phrase-based model (i.e., by augmenting the generated inflected forms in the training corpus) in order to make the proper comparison with morphology injected models.

A factored corpus is created using factor generation tool with factors discussed in Section 6. We generate the morphological inflectional forms for both noun and verb as described in Section 5 with the help of syntactic and morphological tools. We augment the training corpus with these generated inflectional forms and conducted various experiments by changing the factors. We have used various evaluation methods for analyzing the quality of our experiments.

Verb factors experiments

Details of the verb factor models are shown in Figure 6. The models include a phrase-based model and factored models trained with lemma and POS tag as factors.

Modal	Factors	Mapping steps
Factor-based	E: Surface | Lemma | POS H:Surface | Lemma | Suffix | POS	T: 0-0,3
	E:Surface | lemma | POS H:Surface | Lemma | Suffix | POS	D0: T: 0-0, 3 D1: T: 1-1+1-2, 3 G: 1,2,3-0
	E:MainVerbLemma | Tense | Person | Number H:VerbMerged | MainVerbLemma | SuffixMerged	T:0-1+1,2,3-2 G:1,2-0

Figure 6. Factored model mapping to handle verbal inflections

Noun factors experiments

Modal	Factors	Mapping steps
Factor-based	E: Surface | Root | Number | case H:Surface | Root | Suffix	T: 1,2,3-1,2 G: 1,2-0

Figure 7. Factored model mapping to handle Nominal inflections

An English-Hindi dictionary is created that contains all the inflected forms of the noun entities based on number and case factors. Dictionary is merged with the factored corpus. Finally, system is trained using Moses decoder. Training takes around 20-30 minutes. System was tested on 1500 English sentences. A baseline factored system is trained using the same procedure (factored mappings and training data) but without including the dictionary. Details of the Noun factor models are shown in Figure 7.

8.1 Automatic evaluation

Table 8: Automatic evaluation of the translation systems for both Phrase and factor based models

Morph Problem	Model	BLEU Score					
		Without Tuning			With Tuning		
		En -Hi	En -Mr	En-Ml	En -Hi	En -Mr	En -Ml
Noun	Fact	24.30	16.84	24.17	26.30	17.84	25.23
	Fact- Morph	33.41	23.85	33.42	35.41	24.85	35.01
Noun	Phrase	23.87	14.77	26.78	24.87	15.34	27.91
	Phrase- Morph	29.19	21.28	31.30	30.49	23.58	32.72
Verb	Fact	26.03	17.02	26.54	27.51	19.02	28.22
	Fact -Morph	36.16	25.82	35.54	39.89	27.74	37.67
Verb	Phrase	24.78	15.17	26.98	26.87	17.27	27.17
	Phrase- Morph	31.29	23.28	31.41	33.46	25.43	32.76
Noun & Verb	Fact	23.93	15.25	23.01	25.21	16.78	25.65
	Fact- Morph	32.93	22.38	31.56	34.16	23.32	33.16
Noun & Verb	Phrase	25.87	16.37	25.51	27.43	17.62	27.45
	Phrase- Morph	33.19	24.28	32.03	34.65	26.93	34.12

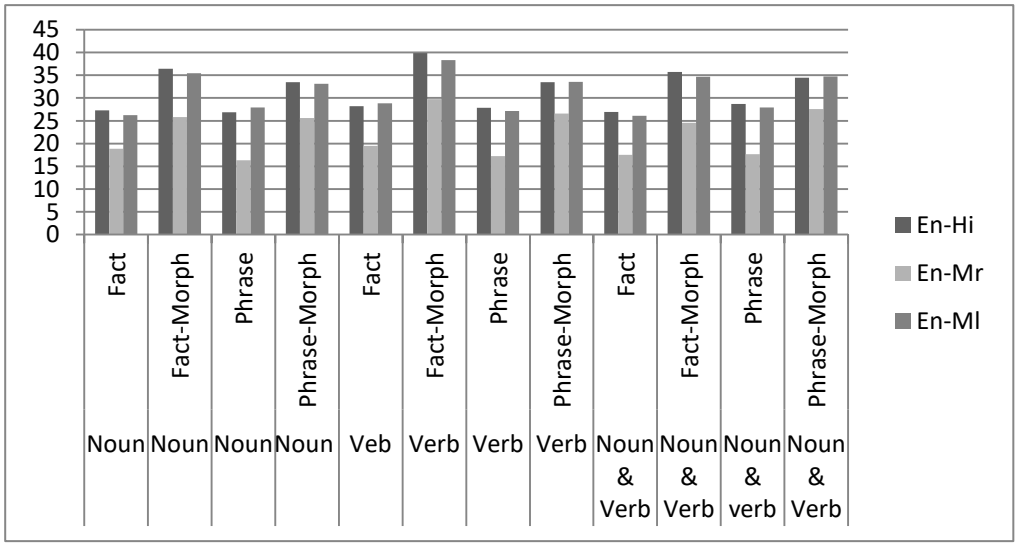


Figure 8: En–Hi, En-Mr, En-Ml BLEU Score Evaluation graph

The translation systems were evaluated using BLEU score [Papineni et al., 2002]. Table 8 shows the BLEU evaluation scores of the translation systems for both Phrase and Factor based models. Figure 8 shows the BLEU score evaluation comparison graphs for the respective En-Hi, En-Mr and En-Ml pairs. From the automatic evaluation scores, it is very evident that Fact-Morph/Phrase-Morph outperforms Fact/Phrase while solving any morphology problem in Hindi, Marathi and Malayalam.

8.1.1. Phrase-based vs. Factor-based models performance comparison

Consider the En-Hi systems, in the case of Noun morphology injection: Fact-Morph shows an improvement of 9.11 BLEU score while Phrase-Morph shows an improvement of 5.62. In the case of Verb morphology injection: Fact-Morph shows an improvement of 12.38 BLEU score while Phrase-Morph shows an improvement of only 6.59. In the case of Noun & Verb morphology injection: Fact-Morph shows an improvement of 8.95 BLEU score while Phrase-Morph shows an improvement of 7.22. Since Phrase-Morph and Fact-Morph have comparative BLEU scores, we have also performed manual evaluation as described in Section 8.2, which showed that fluency and adequacy of the Fact-Morph are in most cases better when compared with Phrase-Morph. The improvement of Fact-Morph is higher when compared with Phrase-Morph. The possible reason may be because in the latter case, we are just injecting morphological forms into the corpus without providing any extra information about when to use them.

For example, *noun {boys}* in English can translate to *लड़के (ladake) or लड़कों (ladakon)* in Hindi. Suppose, we train a phrase based model with the training data having evidence of only boys *लड़के (ladake)*. We also train a factored model as described in Section 5 on the same data but with case as an extra factor. Hence, factored training corpus will have evidence of only *boys|boy|direct लड़के (ladake)|लड़का (ladakaa)|ए (e)*. Now, we inject a word form boys *लड़कों (ladakon) and boys|boy|oblique लड़कों (ladakon)|लड़का (ladakaa)|ओं (on)* in the training corpus of phrase based and factored model, respectively. Then, phrase based model has equal probability to translate boys to *लड़के (ladake) or लड़कों (ladakon)*. This ambiguity may lead to incorrect choice of word while translating. On the other hand, factored model knows when to use which form correctly based on the direct and oblique case.

One important point to note is, Fact-Morph improvement is higher while using Verb/Noun morphology alone compared to the Noun & Verb combined factor model. On the other hand; phrase based models improvement is higher while using Noun & Verb combined models rather than the Verb/Noun alone model. In the case of factor-based approach: the number of translation options increase exponentially with the number of factors. Combination of target factors does not exist in the training data while generating surface form. Finding the correct combination of steps and factors cannot be done easily by brute force. The number of possibilities explodes no matter which direction of exploration it takes. It is difficult to find the correct matching translation with both noun and verb factors. The probability to find target translations with noun and verb factor together is less compared to noun or verb factors separately. Hence there is a reduction in the accuracy to select the best translation, hence the score is less. In the case of phrase-based approach: the alignment options in the trained model have more choices and the probability to find the target translations with inflectional forms for noun and verb together is high. Hence there is an increase in the accuracy to select the best translation, hence the score is high.

Even though, Fact-Morph showed much improvement after morphology injection, phrase based models profit quite a lot from the morphology injection. Since, in the case of Fact-Morph, we have to generate the factors in addition to the morphology generation process. On the other hand, Phrase-Morph systems can work with the generated morphological forms alone and can produce reasonable quality translations. Hence, the time and effort to generate factors is saved in Phrase-Morph.

8.1.2. OOV (Out-Of-Vocabulary) words

Table 9: Counts of total OOV words present before morphology injection and the % OOV words reduction after Morph Injection

Morph Problem	Model	# OOV words			OOV words reduction (%)		
		En-Hi	En-Mr	En-Ml	En-Hi	En-Mr	En-Ml
Noun	Fact	3,030	2,399	2,706	54.14	54.67	58.02
	Fact- Morph	1,739	1,369	1,489			
Noun	Phrase	1567	1945	1856	43.03	37.92	42.85
	Phrase- Morph	1012	1325	1201			
Verb	Fact	3,041	2,772	2,894	81.50	59.85	61.42
	Fact -Morph	1280	1,495	1,534			
Verb	Phrase	1498	1853	1798	49.56	41.11	48.08
	Phrase- Morph	903	1221	1101			
Noun & Verb	Fact	3,393	4,137	4,124	58.02	36.40	55.00
	Fact- Morph	1,867	2,863	2,345			
Noun & Verb	Phrase	1,613	2,172	2,312	61.63	50.37	52.264
	Phrase -Morph	853	1,298	1,354			

We have counted the number of OOV words in the translation outputs, since the reduction in number of unknowns in the translation output indicates better handling of data sparsity. Table 9 shows the OOV words reduction numbers statistics. Figure 9 shows the OOV word reduction comparison graphs for the respective En-Hi, En-Mr and En-Ml pairs. Percentage reduction in OOV shows that, morphology injection is more effective with factored models than with the phrase based model. Also, improvements shown by BLEU are less compared to the percentage reduction in OOV. Since, the ambiguity in lexical choices affects the BLEU improvement. Word form dictionary may have word forms of multiple Hindi or Marathi root words for a single parallel English root word. Hence, often the translation of the English word may not match the reference used for BLEU evaluation, even though it may be very similar in the meaning. Table 10 shows the number of OOVs that are actually translated after morphology injection and the number of translated OOVs that match with the reference. Figure 10 shows the comparison graph for OOVs translated vs. Reference matches for the En-Hi, En-Mr and En-Ml language pairs. We observe that matches with the reference are very low compared to the actual number of OOVs translated. Thus, BLEU score cannot truly reflect the usefulness of morphology injection.

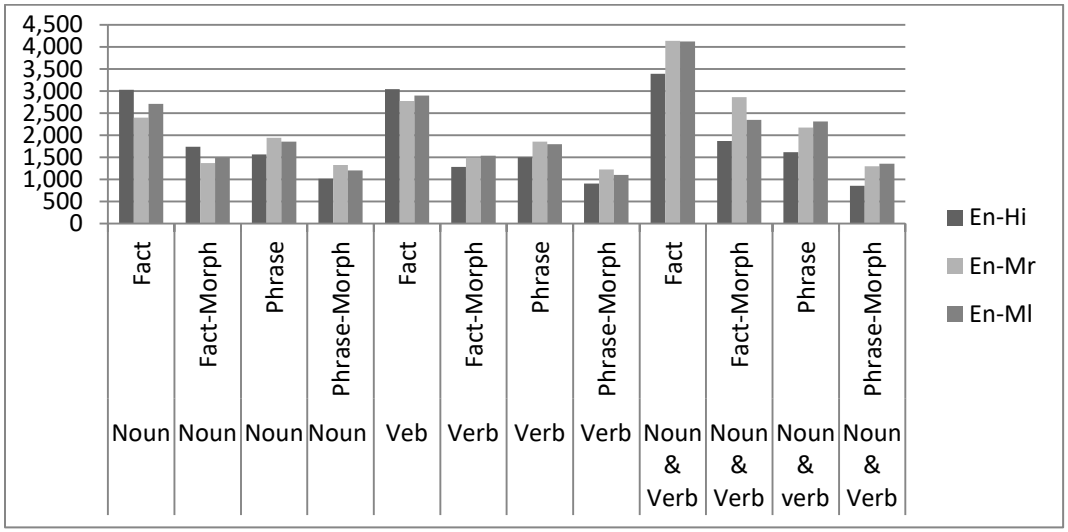


Figure 9: En–Hi, En-Mr, En-Ml OOV word counts Comparison graph

Table 10: Counts of total OOVs translated after morphology injection and the matches with the reference used for BLEU evaluation

Morph Problem	En-Hi		En-Mr		En-Ml	
	# OOV translated	# Ref. Matches found	# OOV translated	# Ref. Matches found	# OOV translated	# Ref. Matches found
Noun (Fact)	1291	558	1030	248	1217	523
Noun (Phrase)	555	123	720	213	655	143
Verb (Fact)	1761	971	1077	253	1360	642
Verb (Phrase)	595	114	832	207	697	152
Noun & Verb (Fact)	1526	687	1174	284	1779	613
Noun & Verb (Phrase)	760	71	674	116	458	123

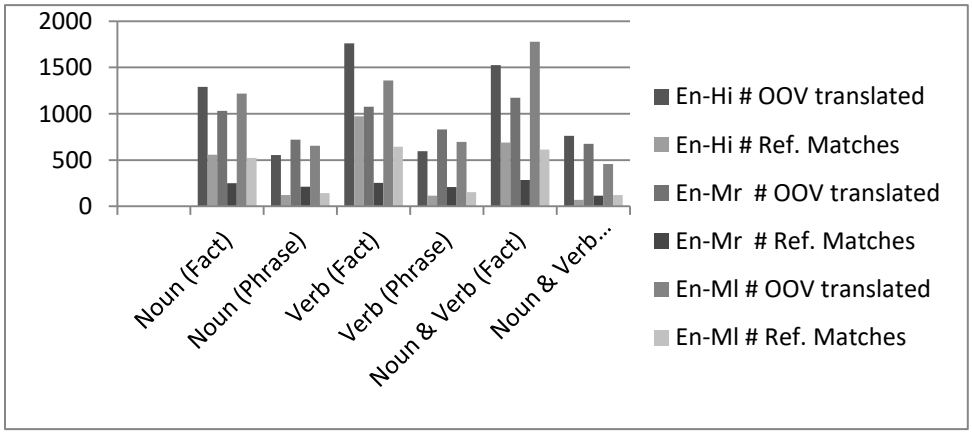


Figure 10: En – Hi, En-Mr, En-Ml total OOVs translated vs. Reference matches Comparison graph

8.2 Subjective (Human) Evaluation

Table 11. Subjective evaluation scheme for Adequacy [Ramananthan et al., 2009]

Score	Level Interpretation
5	All meaning is conveyed
4	Most of the meaning is conveyed
3	Much of the meaning is conveyed
2	Little meaning is conveyed
1	None of the meaning is conveyed

In addition to the impressive improvement in BLEU, we also performed human evaluation. Since, BLEU evaluation with only a single reference is not a true measure for evaluating our method. We found out that Fact-Morph/Phrase-Morph systems give better outputs compared to Fact/Phrase systems in terms of both adequacy and fluency. We randomly chose 250 translation outputs from each system for manual evaluation to get adequacy (A) and fluency (F) scores. The scores were given on the scale of 1 to 5 going from worst to best, respectively. Table 11 and Table 12 shows the evaluation schemes used in [Ramanathan et al., 2009]. The formula used for computing the scores is:

$$A/F = 100 * \frac{(S5 + 0.8 * S4 + 0.6 * S3)}{N}$$

For computation, we considered the sentences (S) with scores above 3 only. In order to make the estimate of scores much better, we penalize the sentences with scores 4 and 3 by multiplying their count with 0.8 and 0.6 respectively. Table 13 shows the average adequacy and fluency scores for each system. We also observe up to 58.87% improvement in adequacy and up to 72.54% improvement in fluency for the English to Hindi systems and up to 49.32% improvement in adequacy and up to 60.78% improvement in fluency for the English to Marathi systems and up to 58.89% improvement in adequacy and up to 71.23% improvement in fluency for the English to Malayalam systems. Figures 11 and 12 shows the adequacy and fluency comparison graphs for the respective En-Hi, En-Mr and En-Ml pairs.

Table 12 : Subjective evaluation scheme for Fluency [Ramananthan et al., 2009]

Score	Level Interpretation
5	Flawless Hindi, with no grammatical errors whatsoever
4	Good Hindi, with a few minor errors in morphology
3	Non native Hindi, with possibly a few minor grammatical errors
2	Disfluent Hindi, with most phrases correct, but ungrammatical overall
1	Incomprehensible

Table 13. Subjective evaluation of the translation systems with and without word form dictionary

Morph Problem	Model	Adequacy			Fluency		
		En- Hi	En - Mr	En - Ml	En - Hi	En - Mr	En-Ml
Noun	Fact	34.32 %	28.01 %	35.12%	38.78 %	32.98 %	35.43%
	Fact-Morph	56.52%	48.41 %	56.32%	65.04%	57.52%	64.32%
Noun	Phrase	33.12%	28.10%	32.54%	34.21%	29.01%	31.34%
	Phrase -Morph	44.87%	38.92%	43.56%	53.23%	48.12%	55.67%
Verb	Fact	37.48 %	30.34%	37.43%	42.72%	38.08%	37.32%
	Fact -Morph	60.87%	49.32%	58.89%	72.54%	60.78%	71.23%
Verb	Phrase	33.89%	28.80%	33.86%	34.98%	30.96%	33.23%
	Phrase- Morph	46.93%	39.87%	48.43%	55.67%	50.56%	57.12%
Noun & Verb	Fact	34.13%	27.56%	34.67%	39.05%	34.01%	38.02%
	Fact- Morph	54.87%	45.45%	52.34%	60.04%	55.32%	61.34%
Noun & Verb	Phrase	34.38%	29.34%	34.17%	36.98%	30.76%	36.12%
	Phrase- Morph	50.96%	40.86%	50.56%	57.43%	54.87%	59.12%

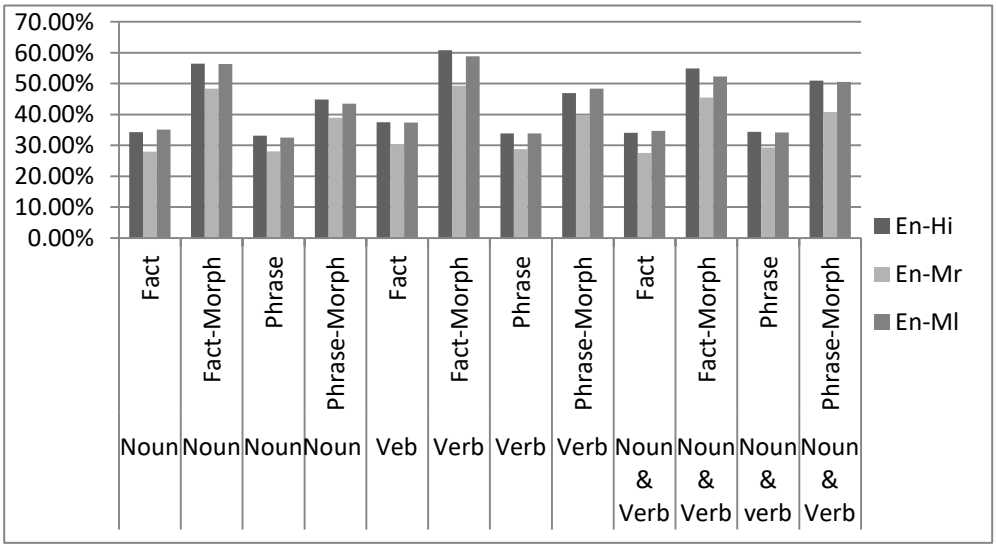


Figure 11: En-Hi, En-Mr, En-Ml Adequacy Comparison graph

For the En-Hi Systems, in the case of Noun morphology injection: Fact-Morph shows an adequacy improvement of 22.2 and a fluency improvement of 28.26; while Phrase-Morph shows an adequacy improvement of 11.75 and a fluency improvement of 19.02. In the case of Verb morphology injection: Fact-Morph shows an adequacy improvement of 23.39 and a fluency improvement of 29.82; while Phrase-Morph shows an adequacy improvement of 13.04 and a fluency improvement of 20.69. In the case of Noun & Verb morphology injection: Fact-Morph shows an adequacy

improvement of 20.74 and a fluency improvement of 20.99; while Phrase-Morph shows an adequacy improvement of 16.58 and a fluency improvement of 20.45. We analyze that, in subjective evaluation also Fact-Morph models outperforms Phrase-Morph systems in all cases in addition to the BLEU score comparison described in Section 8.1.1. Thus, we observed that, the subjective evaluation projects the usefulness of morphology injection in better way compared to BLEU evaluation.

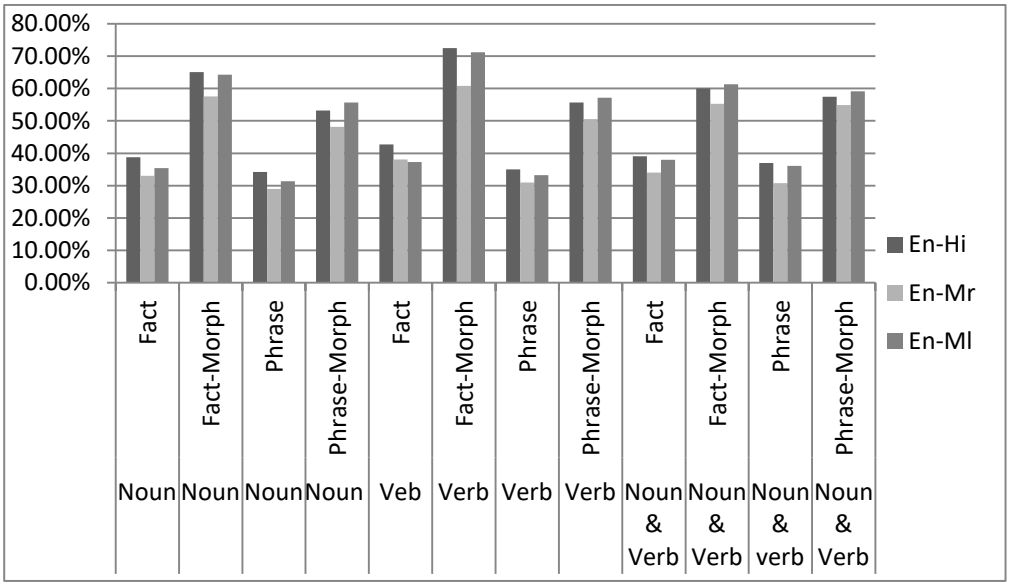


Figure 12: En Hi, En-Mr, En-Ml Fluency Comparison graph

8.3 Error Analysis

We also performed a qualitative evaluation with error analysis. We present some examples in Table 14 with detailed explanation of phenomena with case study. The test cases are taken from the testing corpus and the respective translated outputs of our experiments.

Table 14: Test Cases with examples for English-Hindi translation for various models

Examples	Test Sentences	Explanation of Phenomena
Source Sentence 1	There is a crowd of traders of the world at the auction center. Hindi: वहाँ के नीलम केंद्र पर दुनिया के व्यापारियों का भीड़ है। *{vahan ke nilam kendr par duniya ke vyapariyon ka bheed hae}*	In this case, Fact-Morph correctly translated *auction* to नीलाम *(neelam).* Also, note that, as Fact could not translate *auction*, the next word, *center* is incorrectly translated to मध्य *(madhu)* {middle}. The correct translation is केंद्र *(kendr) (center).* Thus, we also see improvements in the correct lexical choice for the words in local context of the nouns in Fact-
Fact	वहाँ के auction मध्य में दुनिया के व्यापारियों की भीड़ लगी रहती है । *{vahan ke auction madhya mein vyapariyon ki bhiid lagii rahatii hai}* *{ there auction center in trader's crowd is there}*	
Fact-Morph	वहाँ के नीलाम केंद्र में दुनिया के व्यापारियों की भीड़ है । *{vahan ke nilam kendra mein vyapariyon ki bhid lagii rahati hai}*	

	{there in nilam centre world trader's crowd is there}	Morph. On the other hand phrase based system also not able to translate auction center correctly. Moreover it fails to pick the correct word ordering, case markers and verb forms. In the case of Phrase-Morph, the system was able to make the lexical choices correctly but fails in placing the correct word ordering.
Phrase	वहाँ दुनिया के व्यापारियों की भीड़ के auction मध्य में लगी रहती है । *{vahan vyapariyon ki bhid ke auction madhya mein lagi rahti hai}* *{ there world trader's crowd auction center is}*	
Phrase-Morph	वहाँ दुनिया के व्यापारियों की भीड़ नीलाम केंद्र में लगी है । *{vahan uniya ke vyapariyon ki bhiid niilam kendra mein lagii hai}* *{there world trader's crowd is in nilam centre}*	
Source Sentence 2	*Eyelids are a thin fold of skin that cover and protect the eye.* Hindi :पलके जो पतली त्वचा की होती है वो आँखों को डकती और रक्षा करती है । *{palke jo patli tvacha ki hoti hae vo aankhon ko dakti or raksha karti hae}*	In this case, eyelids and fold are not translated by Fact, but Fact-Morph correctly translates them to पलकें (*palkem)* and गुना *(guna),* respectively. On the hand, phrase based system translates *eyelids* to पलक, which misses the inflection form. Also the system fails to translate the verb forms correctly. But the Phrase–Morph system was able to translate the words correctly. Still it fails in handling the proper verb forms.
Fact	eyelids त्वचा की पतली fold हैं कि और आँखों की रक्षा करते हैं *{eyelids tvachaa kii patalii fold hai ki aur aankhon kii rakshaa kartein hain}* *{eyelids skin thin fold and eyes are being protected}*	
Fact – Morph	पलकें त्वचा की पतली गुना हैं कि वो आँखों को डकती और रक्षा करती है *{palaken tvachaa kii patalii gunaa hai ki vo aankhon kii dakti or rakshaa karti hai}* *{palkem skin thin guna and eyes are being covered and protected}*	
Phrase	पलक त्वचा की पतली गुना और आँखों की बचाती *{eyelid tvachaa kii patalii fold aur aankhon kii rakshaa}* *{eyelid skin thin fold and eyes protect}*	
Phrase – Morph	पलकें त्वचा की पतली गुना और आँखों की रक्षा है {paken tvacha ki patli guna or aankhon ki raksha hae} {eyeeelids skin thin guna and eyes are protect}	

9 GENERALIZED SOLUTION

In Section 3, 4 and 5, we have described the sparseness problem and its solution in the context of solving the noun and verb morphology for English as a source language and Hindi as a target language. However, can the process to generate all factor combinations be generalized for other morphologically richer languages on the target side? We have investigated a generalized solution to this problem. We can use technique for new target language X if:

- We identify the factor set, S, that affects the inflections of words in language X and can extract them from English sentences
- We know which inflection the target word will have for a particular combination of factors in S on source side

- We develop a joiner tool in language X to generate the surface word from the root word and suffix.

This is the main objective of the morphology injection process. If we have a source side root word with its factors, we can generate the corresponding inflected form on the target side. As a pre-requisite, we should be aware about the inflectional categories and the information about which factor will generate which inflected form.

For example, consider the inflected form for boy in Hindi:

boy|plural|oblique -> लड़कों *(ladakon)* | लड़का *(ladakaa)* |ओं *(on)*

In this case, when the source word *boy* with *plural* and *oblique* as factors we know that *लड़कों (ladakon)* will be the respective Hindi inflected form for *boy*. Thus, if we know the factors for the source side, we can generate the respective inflected form for the target language.

10 CONCLUSION AND FUTURE WORK

SMT approaches suffer due to data sparsity when translating into a morphologically rich language. We solve this problem by enriching the original data with the missing morphological forms of words. Morphology injection performs very well and improves the translation quality. We observe a huge reduction in number of OOVs and improvement in BLEU score, adequacy and fluency of the translation outputs. Though the approach of solving data sparsity seems simple, the morphology generation may be painful for target languages which are morphologically too complex. Our analysis can be concluded in two ways: In terms of the efforts and time to generate factors; phrase based models with morphology injection is the best alternative. On the other hand, in terms of the generation of accurate machine translation, factor based models with morphology injection can be the best choice by neglecting the effort to generate the factors. A possible future work is to generalize the approach of morphology generation and verify the effectiveness of morphology injection on more morphologically complex languages.

ACKNOWLEDGMENTS

The authors would like to thank Mr. Piyush Dungarwal of IIT Bombay for his supports during this work. The authors would like to thank the anonymous reviewers for their suggestions, which helped to prepare this paper in the present form. The authors would like to thank Department of Science and Technology, Govt. of India for funding under the Women Scientist Scheme (WOS-A) with the project code SR/WOS-A/ET-1075/2014.

REFERENCES

Ananthakrishnan Ramananthan, Pushpak Bhattacharyya, Karthik Visweswariah, Kushal Ladha, and Ankur Gandhe. 2011. *Clause Based Reordering Constraints to Improve Statistical Machine Translation.*IJCNLP.

Anoop Kunchukuttan and Pushpak Bhattacharyya. 2012. Partially modelling word reordering as a sequence labeling problem, COLING 2012.

Anoop Kunchukuttan Abhijit Mishra, Rajen Chatterjee, Ritesh Shah and Pushpak Bhattacharyya, 2014. Shata Anuvadak: Tackling Multiway Translation of Indian Languages, LREC, Rekjyavik, Iceland.

Antony P. J. 2013. *Machine Translation Approaches and Survey for Indian Languages,* The Association for Computational Linguistics and Chinese Language Processing, Vol. 18, No. 1, pp. 47-78

Arafat Ahsan, Prasanth Kolachina, Sudheer Kolachina, Dipti Misra Sharma and Rajeev Sangal. 2010. *Coupling Statistical Machine Translation with Rule based Transfer and Generation*. amta2010.amtaweb.org

Avramidis, Eleftherios, and Philipp Koehn. 2008. Enriching Morphologically Poor Languages for Statis tical Machine Translation. ACL

Birch, A., Osborne, M., and Koehn, P. 2007. CCG Supertags in factored Statistical Machine Translation. In Proceedings of the Second Workshop on Statistical Machine Translation, pages 9–16, Prague, Czech Republic. Association for Computational Linguistics.

Bonnie J. Dorr. 1994. Machine Translation Divergences: A Formal Description and Proposed Solution.Computational Linguistics, 1994.

Carpuat, M. and Wu, D. 2007. Improving Statistical Machine Translation using Word Sense Disambiguation. In Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP CoNLL), pages 61–72, Prague, Czech Republic

Chahuneau, Victor, Eva Schlinger, Noah A. Smith, and Chris Dyer. 2013. *Translating into Morphologically Rich Languages with Synthetic Phrases*. Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics.

De Marneffe, Marie Catherine, and Christopher D. Manning. 2008. *Stanford typed de pendencies manual*. URL http://nlp. stanford. edu/software/dependencies manual. pdf.

Durgar El Kahlout, i. and Oflazer, K. 2006. Initial explorations in English to Turkish statistical machine translation. In Proceedings on the Workshop on Statistical Machine Translation, pages 7–14, New York City. Association for Computational Linguistics.

Eleftherios Avramidis and Philip Koehn. 2008. *Enriching Morphologically Poor Languagesfor Statistical Machine Translation.* Proceedings of Association for Computational Linguistics-08: HLT, pages 763–770, Columbus, Ohio, USA.

Franz Josef Och and Hermann Ney. 2003. *A Systematic Comparison of Various Statistical Alignment Models.* Computational Linguistics,

Franz Josef Och and Hermann Ney. 2001. *Statistical Multi Source Translation.* MT Summit.

Federico, Marcello, Bertoldi, Nicola and Cettolo, Mauro. 2007. IRSTLM: an Open Source Toolkit for Handling Large Scale Language Models. FBK-irst - Ricerca Scientifica e Tecnologica Via Sommarive 18, Povo (TN), Italy

Ganesh Bhosale, Subodh Kembhavi, Archana Amberkar, Supriya Mhatre, Lata Popale and Pushpak Bhattacharyya. 2011. *Processing of Participle* (*Krudanta*) *in Marathi*. ICON.

Gandhe, Ankur, Rashmi Gangadharaiah, Karthik Visweswariah, and Ananthakrishnan Ramanathan. 2011. *Handling verb phrase morphology in highly inflected Indian languages for Machine Translation*. IJCNLP.

Habash, N., Gabbard, R., Rambow, O., Kulick, S., and Marcus, M. 2007. Determining case in Arabic: Learning complex linguistic behavior requires complex linguistic features. In Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP CoNLL), pages 1084–1092.

Habash, N. and Sadat, F. 2006. Arabic preprocessing schemes for statistical machine translation. In Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers, pages 49–52, New York City, USA. Association for Computational Linguistics.

Huang, L., Knight, K., and Joshi, A. 2006. Statistical syntax directed translation with extended domain of locality. Proc. AMTA, pages 66–73.

Kevin Knight. 1999. Decoding complexity in word replacement translation models, Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward and Wei Jing Zhu. 2002. *BLEU: a Method for Automatic Evaluation of Machine Translation*, Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL), Philadelphia, pp. 311 318.

Koehn, Philipp. 2005. Europarl: A parallel corpus for statistical machine translation. MT Summit.

Koehn, Philipp, Franz Josef Och, and Daniel Marcu. 2007. *Statistical phrase based translation*. Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology Volume 1. ACL.

Koehn, Philipp and Hieu Hoang. 2007. *Factored Translation Models*. In Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP CoNLL), pages 868–876.

Koehn, Philipp, Hieu Hoang, Alexandra Birch, Chris Callison Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra

Constantin, Evan Herbst. 2007. *Moses: Open Source Toolkit for Statistical Machine Translation*, Annual Meeting of the ACL, demonstration session, Prague, Czech Republic.

Marton, Y., Callison Burch, C. and Resnik, P. 2009. *Improved Statistical Machine Translation Using Monolingually derived Paraphrases*, Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing(EMNLP), Volume 1 Pages 381 390.

Minkov, E., Toutanova, K., and Suzuki, H. 2007. Generating complex morphology for machine translation. In ACL 07: Proceedings of the 45th Annual Meeting of the Association of Computational linguistics, pages 128–135, Prague, Czech Republic. Association for Computational Linguistics.

Nakov, P. I. and Ng, H. T. 2012. Improving Statistical Mahcine Translation for a Resource Poor Language Using Related Resource Rich Languages, Journal of AI Research, Volume 44, pages 179 222.

Peter E Brown, Stephen A. Della Pietra. Vincent J. Della Pietra, and Robert L. Mercer*. 1993. The Mathematics of Statistical Machine Translation: Parameter Estimationn. *ACL*.

Ramanathan, Ananthakrishnan, Hansraj Choudhary, Avishek Ghosh, and Pushpak Bhattacharyya. 2009. *Case markers and morphology: addressing the crux of the fluency problem in English-Hindi SMT*. Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2 Volume 2. Association for Computational Linguistics.

Singh, Smriti, Vaijayanthi M. Sarma. And Stefan Muller. 2010. Hnid Noun Inflection and Distributed Morphology. Universite Paris Diderot, Paris 7, France. Stefan Muller(Editor) CSLI Publications http:.. csli publications.stanford.edu(2006):307.

Singh, Smriti, Vaijayanthi M. Sarma. 2011. Verbal Inflection in Hindi: A Distributed Morphology Approach. PACLIC.

Sreelekha. S, Piyush Dungarwal, Pushpak Bhattacharyya, Malathi.D, 2015. Solving Data Sparsity by Morphology Injection in Factored SMT, International Conference on Natural Language Processing-ICON.

Sreelekha, Pushpak Bhattacharyya, Malathi D. 2014. *Lexical Resources for Hindi –Marathi MT*, WILDRE Proceedings, LREC.

Sreelekha, Pushpak Bhattacharyya. 2016. *Lexical Resources to enrich English Malayalam Machine Translation*, LREC –International Conference on Lexical Resources and Evaluation, Slovenia.

Sreelekha. S, Pushpak Bhattacharyya, Malathi.D. 2017. "Statistical vs. Rule Based; A Case Study on Indian Language Perspective", Springer Journal of Advances in Intelligence and Soft Computing.

Sreelekha. S, Pushpak Bhattacharyya, Malathi.D. 2016. "A Survey Report of Evolution of Machine Translation", International Journal of Control Theory and Applications, IJCTA-9(33), 233-240.

Sreelekha. S, Pushpak Bhattacharyya. 2015. "A Case study on English Malayalam Machine Translation" iDravidian Proceedings.

Sreelekha, Raj Dabre, Pushpak Bhattacharyya, 2013. "*Comparison of SMT and RBMT, The Requirement of Hybridization for Marathi – Hindi MT"* ICON, 10th International conference on NLP.

Shachi Dave, Jignashu Parikh and Pushpak Bhattacharyya. 2002. "Interlingua based English-Hindi Machine Translation and Language Divergence", JMT.

Sunil R, Nimtha Manohar, Jayan V, KG Sulochana. 2011, "*Development of Malayalam Text Generator for translation from English"*, India Conference (INDICON), Annual IEEE.

Tamchyna, Ales¨, and Ond¨rej Bojar. 2013. "*No free lunch in factored phrase based machine translation*. Computational Linguistics and Intelligent Text Processing". Springer Berlin Heidelberg. 210-223.

Toutanova, Kristina, Dan Klein, Christopher D. Man ning, and Yoram Singer, 2003. "*Feature rich part of speech tagging with a cyclic dependency network"*. Proceedings of the 2003 Conference of the North American Chapter of the Association for Computa tional Linguistics on Human Language Technology Volume 1. Association for Computational Linguis tics.

Ueffing, N. and Ney, H. 2003. "Using pos information for statistical machine translation into morphologically rich languages". In EACL '03: Proceedings of the tenth conference on European chapter of the Association for Computational Linguistics, pages 347–354, Morristown, NJ, USA. Association for Computational Linguistics.

Yamada, Kenji. And Knight, Kevin. 2001. "*A Syntax based Statistical Translation Model"*. Proceedings of the 39th Annual Meeting on Association for Computational Linguistics.