Chapter 1

Hierarchical phrase based Machine Translation: Literature survey

In this chapter, we provide a brief overview of machine translation in general and hierarchical phrase based machine translation in particular.

1.1 Machine translation

Machine Translation has its roots in cold war which led Russian to English translation. But even after war was over, US government continued its effort in this field. But the research went in vain, when Automatic Language Processing Advisory Committee (ALPAC) report (1966) exposed that the MT project had hardly fulfilled the promises it made ten years back. In the 80s, this field again started to blossom when the computing power of machines had increased. This period was marked by the introduction of very exciting statistical models for MT.

1.1.1 Approaches

Machine translation is linguistically motivated because it aims at achieving the most appropriate translation from one language to other. This means that a MT system will attain success only after it attains natural language understanding. Generally speaking, rule-based approaches involve an intermediary symbolic language obtained from the source language. This intermediate language is translated to the foreign language. Depending upon how the intermediary symbolic language is obtained, an approach is categorized as Transfer-based machine translation or Interlingua based machine translation. These methods require extensive resources and annotated training set along with large number of rules.

Rule-based MT

Rule-based techniques are linguistically driven methods of MT in the sense that they require dictionary and grammar to understand the syntactic, semantic and morphological aspects of both languages. The main approach of these methods is to obtain the shortest path from one language to another using rules of grammar. Two approaches of rule-based MT are based on interlingua and transfer-based MT. Transfer-based machine translation is based on the idea of interlingua.

Interlingual MT

Interlingua is an intermediate symbolic language that captures the meaning of the sentence in source language, sufficient to convert that into target language. This intermediate symbolic language has no dependence on either source or target language while in transfer-based MT, the interlingua obtained is somewhat dependent on the language pair. The prime reason to go for interlingua is that if there are n languages, we need only 2n translation models instead of $\binom{n}{2}$. Each language is converted into the interlingua that contains the syntax, semantic and morphology and then the interlingua can be converted to any of the language. Another advantage is that people can develop the decoders and encoders independent of the source language. For example, for Chinese to Hindi translation and vice versa, Chinese to Interlingua decoder is programmed by scientist X who has no knowledge about Hindi language. Same goes for scientist Y who is developing Interlingua to Hindi decoder.

Dictionary-based MT

This approach refers to the usage of a dictionary to translate the sentence word-by-word without caring much about the context. It is the most simple of all MT systems. This system might be used to translate phrases for inventories or catalogs of products and services.

Statistical Machine Translation (SMT)

Statistical machine translation is based on statistical data calculated from parallel corpora. Examples of parallel corpora are Canadian Hansard corpus, the English-

French record of the Canadian parliament. The idea is that if a word pair translation is more frequent in the training data, it is likely that this translation will get a better probability. The entire process works on the basic idea of counting and giving probability to each translation to evaluate the correctness of the translation.

Example-based Machine Translation (EBMT)

In this method, the idea of using statistical data from a parallel corpora is extended to the next level. The system looks for similar patterns that exist in the training data and gives a translation based on examples from the training data. The first EBMT system was developed by Nagao [1984] in 1984.

Hybrid Machine Translation

As the name suggests, it takes advantage of both rule-based and statistical approaches to devise a better translation technique. One approach is to obtain the translation using rule-based MT and then correct the translation using a statistical MT.

1.1.2 Major Issues in Machine Translation

In this part, we discuss some of the frequently encountered problems in MT.

Word sense disambiguation (WSD)

A word can have several senses. For example, bank can either mean riverbank or a financial institution. WSD tries to disambiguate the sense of the word either using shallow or deep techniques. Shallow techniques assume no previous knowledge about the word, but use statistics concerning the word sense by looking at neighboring words. Deep techniques have knowledge about the various senses of the word. Despite the knowledge backup, shallow techniques perform better compared to deep techniques.

Named entity recognition

Nouns come in different forms like persons, organizations, locations, expressions of times, quantities, monetary values, percentages, etc. The job of a Named Entity Recognizer (NER) is to correctly classify nouns into one of these categories.

Although the job of a NER seems trivial, it has been observed that the best rule-based and statistical implementation of NER performs poorly in domains other than the one they are trained in. This has made the development of a universal NER mandatory. In the next section, we discuss phrase based machine translation model.

1.1.3 Phrase based model

Phrase-Based models (Koehn et al. [2003]) advanced the previous machine translation methods by generalizing translation. Earlier, the words were considered as a basic unit of translation. Phrase-Based methods introduced phrases as a basic unit of translation. So sentences were concatenation of two or more phrases. This approach is good at removal of translation error caused due to local reordering, translation of short idioms, insertions and deletions.

Noisy channel approach

Basic phrase-based model is an instance of the noisy channel approach (Brown et al. [1990]). The translation of a french sentence f into an English sentence e is modeled as:

$$\underset{e}{\operatorname{argmax}} P(e|f) = \underset{e}{\operatorname{argmax}} P(e) * P(f|e) \tag{1.1.1}$$

The translation model

1. Segment e into phrases $\bar{\text{e}}_1 \ldots \bar{\text{e}}_n$;

2. Reorder the $\bar{\text{e}}_i$'s according to some distortion model;

3. Translate each of the $\bar{\text{e}}_i$ into French phrases according to a model $P(\bar{\text{f}}|\bar{\text{e}})$ estimated from the training data.

Other phrase-based models

There are other phrase-based models such as the joint distribution P(e,f) or the one that makes P(e) or P(f|e) as features of log-linear model. Despite this fact the basic architecture consists of the same building blocks like phrase segmentation or generation, phrase reordering and phrase translation.

Salient features of a phrase-based model

Phrase-Based models are very good in performing translations at the phrase level that have been observed from the training data. The performance of translation hardly improves as the length of substring increases beyond three words because this method relies heavily on training data. So it fails to handle sparseness of data and provide translation for longer phrases. The distortion algorithm works on top of phrase model and reorders phrase irrespective of the words in their neighborhood.

Drawbacks of phrase-based models

Often it is required to capture translations that are relevant beyond the standard three word phrase. As an example, we consider a Chinese to English translation followed by an Odia to English translation and show how phrase-based translation cannot translate longer phrases and we need special structures.

A word by word translation

First we obtain a word by word translation for each language pair.

Chinese to English Aozhou$_1$ shi$_2$ yu$_3$ Bei$_4$ Han$_5$ you$_6$ bangjiao$_7$ de$_8$ shaoshu$_9$ guojia$_{10}$ zhiyi$_{11}$.
Australia$_1$ is$_2$ with$_3$ North$_4$ Korea$_5$ have$_6$ diplomatic$_7$ relations$_7$ that$_9$ few$_{10}$ countries$_{11}$ one$_{12}$ of$_{13}$.

Odia to English Australia$_1$ tee$_2$ alpa$_3$ desh$_4$ madhiyare$_5$ gotiye$_6$ emiti$_7$ jahar$_8$ uttar$_9$ korea$_{10}$ sangare$_{11}$ rajnaik$_{12}$ sampark$_{13}$ achi$_{14}$.
Australia$_1$ is$_2$ few$_3$ countries$_4$ of$_5$ one$_6$ that$_8$ Northr$_9$ Korea$_{10}$ with$_{11}$ diplomatic$_{12}$ relations$_{13}$ have$_{14}$.

1.1.4 Problem with Phrase based MT

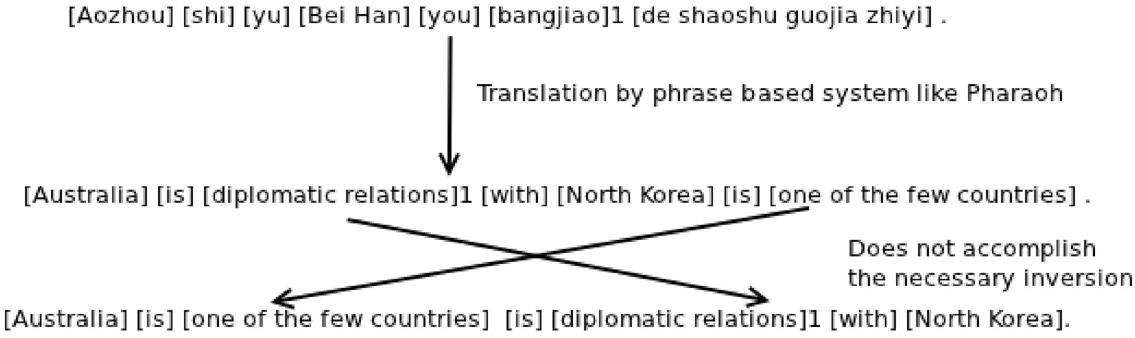

[Aozhou] [shi] [yu] [Bei Han] [you] [bangjiao]1 [de shaoshu guojia zhiyi] .
Translation by phrase based system like Pharaoh
[Australia] [is] [diplomatic relations]1 [with] [North Korea] [is] [one of the few countries] .
Does not accomplish the necessary inversion
[Australia] [is] [one of the few countries] [is] [diplomatic relations]1 [with] [North Korea].

Figure 1.1: Chinese to English phrase-based translation

When we ran phrase-based MT systems like Pharaoh on the Chinese sentence, we got the second sentence. Although it correctly translates "diplomatic relations with North Korea" and "one of the few countries", it is not able to apply the necessary inversion of those two groups. Some other complicated reordering models like the lexical phrase reordering model might be able to accomplish such inversions, simpler distortion models will inevitably fail. The problem is not in the distortion model, but in identifying basic units of translation as we will discuss in Chapter 1.2.

1.2 Hierarchical phrase based MT

In phrase based MT, the basic unit of translation is phrase. Hierarchical model brings sub-phrases into existence to remove the problems associated with phrase-based MT. Let us see an English to Hindi example. Consider the translation in Figure 1.2. We reduce this observation into a grammatical rule. A possible grammar rule is that the phrases on either side of the word *of* will be swapped when translating to Hindi. This is the advantage of using sub-phrases. In case of phrase level translation, this rotation is fixed only for a particular phrase and there are different rules for other phrases requiring similar rotation. This contributes to increasing redundant rules. We give some examples of phrase based translation to understand how redundancy is introduced in 2.1

In phrase based MT, these redundant rules are stored in a dictionary. On the contrary, hierarchical machine translation replaces these rules by a single rule i.e.

$$X \rightarrow \langle X_1 \text{ का } X_2 , X_2 \text{ of } X_1 \rangle$$

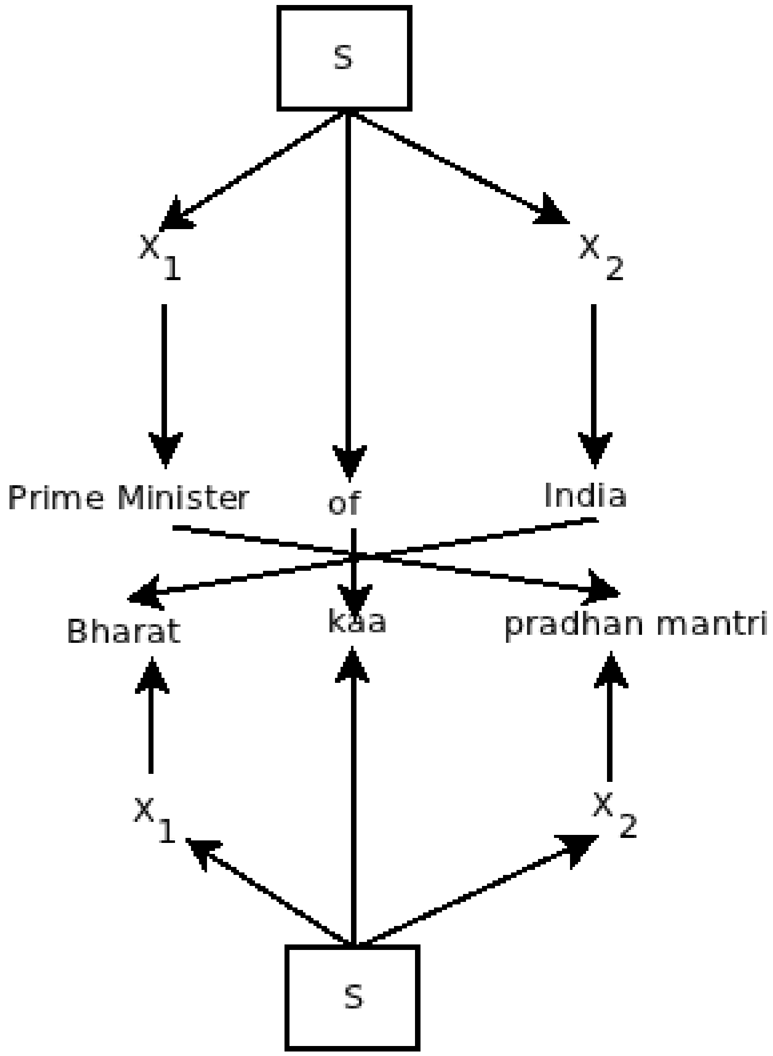


Figure 1.2: Hindi to English translation showing reordering

Every rule is associated with a weight w that expresses how probable the rule is in comparison to other rules with same rule in the Hindi side.
For ex:- भारत का राष्ट्रीय पक्षी {bhaarata kaa raastriiya pakshii} {India of National bird} → National bird of India bird
This example will have a similar expression on the Hindi side but different on the English side.

$$X \rightarrow \langle\ X_1 \text{ का } X_2, X_1\ '\ s\ X_2\ \rangle$$

Note that the ordering remains same.

Basically, hierarchical model not only reduces the size of a grammar, but also combines the strength of a rule-based and a phrase-based machine translation system. This can be observed from the working of grammar extraction or decoding because hierarchical model uses rules to express longer phrases and phrases as it is for smaller phrases.

The grammar used for translation is very interesting in the sense that the system requires the same rules for parsing as well as translation. This kind of grammar is formally called synchronous context free grammar. Synchronization is required between sub-phrases because these sub-phrases need to have a number attached to them since they are essentially all X. X is the only symbol used as a non-terminal apart from the start state S. The numbering system is the way non-terminals are differentiated.

This model does not require parser at the Hindi side because all phrase are labelled as X. This is very important with respect to Indian languages, since none of the Indian languages have a good automated parser at the moment.

Phrase based systems are good at learning reordering of words. So the hierarchical model uses phrase based reordering technique to learn reordering of phrases. This can be achieved if the basic units of translation are combination of phrases and words. Systems using hierarchical models emphasize on the hypothesis that hierarchy may be implicit in the structure of a language. In the following sections, we demonstrate some grammar rules that can be automatically extracted from corpus.

Phrases are good for learning local reordering, translations of multi-word expressions, or deletion and insertions that are sensitive to local context. As we have seen in previous examples, a phrase based system can perform reordering with phrases that were present during training, but if it comes across unknown phrases that were actually not there in the corpus but are similar to a rule observed from the corpus, it will not provide the correct translation. This has been illustrated in 2.2

1.3 Summarising the defects in phrase based model compared to hierarchical phrase based model

Phrase based models can perform well for translations that are localized to substrings and have been observed previously in the training corpus. Also learning phrases longer than three words hardly improves the performance because such phrases may be infrequent in the corpus due to data sparsity. The natural way seems to be learning small phrases and some grammatical rules and combining them to produce a translation.

There are also phrase based systems that try to introduce reordering termed as distortion independent of their content. But this is like fighting with your opponent blindfolded. Every reordering should be accompanied by the use of context.

All these problems are handled well by hierarchical phrase model. Certainly a leap above phrase based model, because hierarchical phrases can contain sub-phrases allowing for natural rotation of sub-phrases and learning of grammar rules.

The system learns these rules from parallel corpus without any syntactic annotation that is essential for Indian to English language MT (IELMT). The system adopts technology from syntax based machine translation system but includes the flavor of hierarchical phrases thus presenting a challenging problem.

1.4 Some notes about the system

The system that we describe later will be using rules called transfer rules. It learns such rules automatically from an unannotated bitext. Thus, this system does not require any kind of syntactic knowledge from the training data.

1.4.1 Synchronous context free grammar

Synchronous context free grammar is a kind of context free grammar that generates pair of strings.

Example:- $S \rightarrow I$,मेन

This rule translates 'I' in English to मेन{main} in Hindi. This rule consists of terminals only i.e., words but rules may consist of terminals and non-terminals as described below.

$$VP \rightarrow \langle\ V_1\ NP_2, NP_2\ V_1\ \rangle$$

Use of synchronous CFG

The hierarchical phrase pairs can be seen as synchronous CFG. One might say that this approach is similar to syntax based MT. This is not true because the hierarchical phrase based MT system is trained on a parallel text without making any linguistic assumption that the data is annotated with part-of-speech.

Demonstrative Example

$$\begin{array}{ll}
S \rightarrow \langle\ NP_1\ VP_2, NP_1\ VP_2\rangle & (1) \\
VP \rightarrow \langle\ V_1\ NP_2, NP_2\ V_1\ \rangle & (2) \\
NP \rightarrow \langle\ \text{i, watashi wa}\ \rangle & (3) \\
NP \rightarrow \langle\ \text{the box, hako wo}\ \rangle & (4) \\
NP \rightarrow \langle\ \text{open, akemasu}\ \rangle & (5)
\end{array}$$

How does this grammar work?

The parse tree begins with a start symbol in CFG but in synchronous CFG parser starts with a pair of start symbols.

Example:- $\langle\ S_{10}, S_{10}\ \rangle$

This rule means there are two parse trees instead of one. We number this symbols to avoid ambiguities when there are same elements (non terminals) occurring twice on both sides.

Example:- $\langle\ NP_{11}\ V_{13}\ NP_{14}, NP_{11}\ NP_{14}\ V_{13}\ \rangle$

Here we see that two NP symbols are co-occurring on the same side. If they are not indexed, there can be ambiguity over the correspondence of a non-terminal on the target side. This ambiguity is resolved by indexing the symbols. In this way, the non terminals are synchronized and hence this grammar is called synchronous grammar.

Next we substitute the rule for S based on the grammar.

$\langle\ NP_{11}\ V_{12},\ NP_{11}\ VP_{12}\ \rangle$
$\Rightarrow \langle\ NP_{11}\ V_{13}\ NP_{14},\ NP_{11}\ NP_{14}\ V_{13}\ \rangle$
$\Rightarrow \langle$ i $V_{13}\ NP_{14},\ NP_{11}$ watashi wa $V_{13}\ \rangle$ (not allowed)
$\Rightarrow \langle$ i $V_{13}\ NP_{14}$, watashi wa $\mathrm{NP}_{14}\ \mathrm{V}_{13}\ \rangle$
$\Rightarrow \langle$ i open NP_{14}, watashi wa NP_{14} akemasu$\rangle$
$\Rightarrow \langle$ i open the box, watashi wa hako wo akemasu$\rangle$

CFGs as pair of trees

The rules of synchronous CFG can be described as a pair of parse trees. The left hand side rules inside the rule region collectively gives grammar rules for obtaining a parse tree in english language. Consider following examples.

$S\rightarrow \langle\ NP_1\ VP_2\ \rangle$
$VP\rightarrow \langle\ V_1\ NP_2\ \rangle$
$NP\rightarrow \langle\ i\ \rangle$ (not allowed)
$NP\rightarrow \langle$ the box $\rangle$
$V\rightarrow \langle$ open$\rangle$

The parse trees look like in Fig1.3:

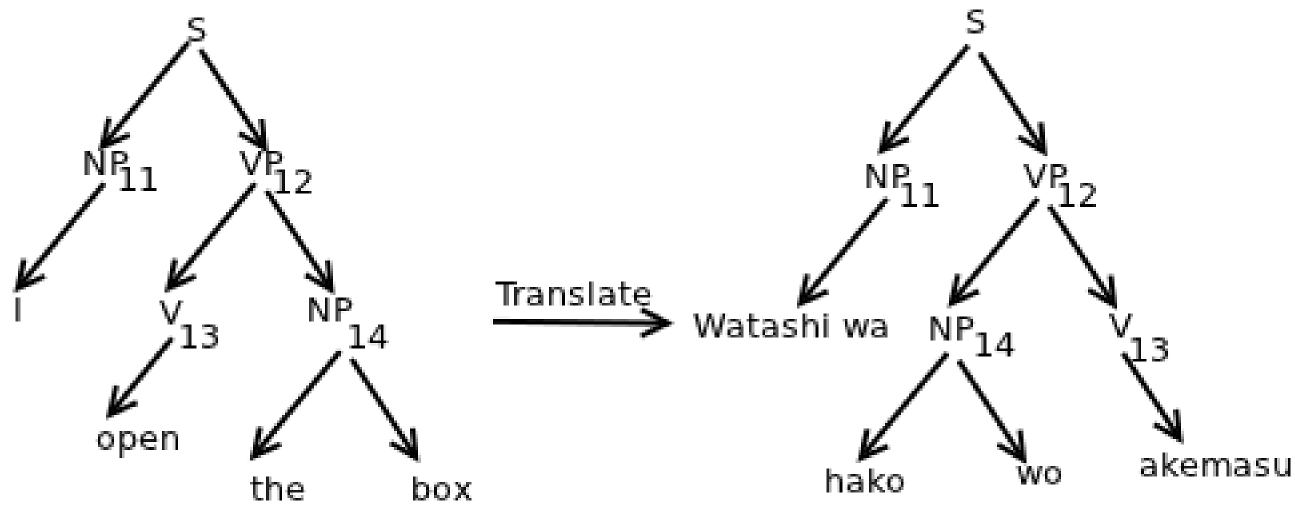


Figure 1.3: Parse tree for translation from English to Japanese

Once we have the parse tree in one language, we can construct the parse tree in other language. To accomplish the construction of the parse tree in target side, we need to apply the transfer rules and obtain the parse tree in the target language. In case there is reordering, the transfer rules cause the terminals or non terminals to rotate about a non terminal which has a corresponding rule in grammar for reordering. This has been demonstrated by the substitutions shown earlier.

1.4.2 The model

The system makes a departure from noisy channel approach to the more general log-linear model.

Log-linear model

The system evaluates a set of features for each rule it derives from the training data. Then it calculates the weight for each feature and obtains product to find the weight-age of each rule of the format X $\rightarrow \langle\gamma, \alpha\rangle$ according to this formula.

$$w(X \rightarrow \langle\gamma, \alpha\rangle) = \prod_i \phi_i(X \rightarrow \langle\gamma, \alpha\rangle)^{\lambda_i} \tag{1.4.1}$$

Note:- ϕ_i are the features and λ_i are the weights given to each feature.

There are five features similar to the ones found in Pharaoh's feature set. The features are :-

1. P($\gamma|\alpha$) and P($\alpha|\gamma$)
2. P$_w$ ($\gamma|\alpha$) and P$_w$ ($\alpha|\gamma$)
3. Phrase penalty

The feature have been divided in three sets in the manner in which they are evaluated.

Feature pair #1

$$P(\gamma|\alpha) = \frac{count(\gamma, \alpha)}{count(\alpha)} \tag{1.4.2}$$

$$P(\alpha|\gamma) = \frac{count(\gamma, \alpha)}{count(\gamma)} \tag{1.4.3}$$

The count of co-occurrences of phrase γ and α can be easily obtained from bitext simultaneously to obtain the probability. The former feature is found in noisy channel model but the latter feature was also found useful to obtain the alignment matrix discussed latter.

Lexical weights

P$_w$ ($\gamma|\alpha$) and P$_w$ ($\alpha|\gamma$) are features which estimate how well the words in phrase γ translate the words in phrase α Koehn et al. [2003].
w($\gamma|\alpha$) - probability distribution for lexical translation.

$$w(\gamma|\alpha) = \frac{count(\gamma, \alpha)}{count(\alpha)} \tag{1.4.4}$$

Given a phrase pair $\langle\gamma, \alpha\rangle$ and a word alignment *a* between the foreign word positions i = 1...n and the English word positions j = 0,1...m, the lexical weight P$_w$ is computed by

$$\prod_{i=1}^{n} \frac{1}{|\{j|(i,j) \in a\}|} . \sum_{\forall (i,j) \in a} w(\gamma_i|\alpha_j) \tag{1.4.5}$$

Consider an example of translation of French phrase *f* and English phrase *e*, the alignment matrix is given as :

	f$_1$	**f**$_2$	**f**$_3$
Null	–	–	##
e$_1$	##	–	–
e$_2$	–	##	–
e$_3$	–	##	–

Table 1.4.1: Alignment matrix.

The alignment matrix provides the one to one mapping by filling the matrix with double hash for an alignment and double blank for non alignment. Based on the alignments and formula suggested above by Koehn, we obtain probability for translation of English phrase *e* to French phrase *f* given alignment *a* as in equation 1.4.6.

$$p_w(\bar{f}|\bar{e}, a) = p_w(f_1 f_2 f_3|e_1 e_2 e_3, a) = w(f_1|e_1) \times \frac{1}{2}(w(f_2|e_2) + w(f_2|e_3)) \times w(f_3|NULL) \tag{1.4.6}$$

Similarly we can obtain the probability in the opposite direction.

Phrase penalty

This feature is also similar to Koehn's phrase penalty which gives the model some flexibility in giving preference to shorter or longer derivations.

Final weight

Then the weight of D is the product of the weights of the rules used in the translation, multiplied by the following extra factors:

$$w(D) = \prod_{\langle r,i,j\rangle \in D} w(r) \times p_{lm}(e)^{\lambda_{lm}} \times exp(\lambda_{wp}|e|) \tag{1.4.7}$$

Where p_{lm} is the language model and $exp(\lambda_{wp}|e|)$, the word penalty gives some control over the length of the english output.

1.5 Decoding

Basically the decoder is a CKY parser with beam search for mapping French derivations to English derivations.
Given a French sentence f, it finds the English yield of the single best derivation that has French yield f:

$$\hat{e} = \underset{D\ s.t\ f(D)=f}{\operatorname{argmax}}\ P(D) \tag{1.5.1}$$

This may not be the highest probability English string, which would require more expensive summation over derivations.
Over the next few sections I discuss the challenging technique to find the probability of single best English translation and the intricacies of decoder.

1.6 Basic Algorithm

A parser in this notation defines a space of weighted items, in which some items are designated axioms and some items are designated goals (the items to be proven), and a set of inference rules of the form

$$\frac{I_1 : w_1 ... I_k : w_k}{I : w}\phi \tag{1.6.1}$$

Which means that if all the items I$_i$ (called the antecedents) are provable, with weight w$_i$, then I (called the consequent) is provable with weight w, provided the condition ϕ holds.
In our previous example:

I$_1$(X→ भारत, India)	: w$_1$
I$_2$(X→ प्रधान मन्त्रि, Prime Minister)	: w$_2$
I$_3$(X→ X_1 का X_2, X_2 of X_1)	: w$_3$

$$\frac{I_1 : w_1\ I_2 : w_2\ I_3 : w_3}{I : w_1 w_2 w_3} \tag{1.6.2}$$

Here is the derivation

I(*भारत का प्रधान मन्त्री* → *Prime Minister of India*)

More formally the well known CKY algorithm for CFGs in CNF can be thought of as a deductive proof system whose items can take one of two forms:

- $[X, i, j]$, indicating that a sub-tree rooted in X has been recognized spanning from i to j(that is spanning f_{i+1}^{j})
- X $\rightarrow \gamma$, if a rule X $\rightarrow \gamma$ belongs to the grammar G.

The axioms would be

$$\frac{}{X \rightarrow \gamma : w}(X \rightarrow \gamma) \in G \tag{1.6.3}$$

And the inference rules would be

$$\frac{Z \rightarrow f_{i+1} : w}{[z, i, i+1]} : w \tag{1.6.4}$$

$$\frac{Z \rightarrow XY : w \;\; [X, i, k] : w_1 \;\; [Y, k, j] : w_2}{[Z, i, j] : w_1 w_2 w3} \tag{1.6.5}$$

And the goal would be $[S, 0, n]$, where S is the start symbol of the grammar and n is the length of the input string f. Given a synchronous CFG, we could convert its French side grammar into Chomsky normal form, and then for each sentence, we could find the best parse using CKY. Then it would be a straight-forward matter to revert the best parse from Chomsky normal form into the original form and map it into its corresponding English tree,whose yield is the output translation. However, because we have already restricted the number of non-terminal symbols in our rules to two, it is more convenient to use a modified CKY algorithm that operates on our grammar directly, without any conversion to Chomsky normal form. Converting a CFG to CNF makes the grammar exponentially bigger, so it is better to keep the grammar, which is already a million lines as a CFG. In the next section, the above technique to transfer a tree to a string has been demonstrated with an Odia - English translation example. The section describes how to obtain grammar rules from a parallel corpus, *i.e.* training, then generating a tree for the Odia sentence, *i.e.* parsing, converting the tree in Odia to a tree in English, i.e. decoding and finally obtaining the yield of the tree in English, which is the translation.

1.7 Training

So far we have obtained a general idea about synchronous context free grammars and its usage. In the following section, we will explain the method deployed to obtain such grammar rules from a parallel corpora or bitext.

1.7.1 Illustration of word alignment algorithm

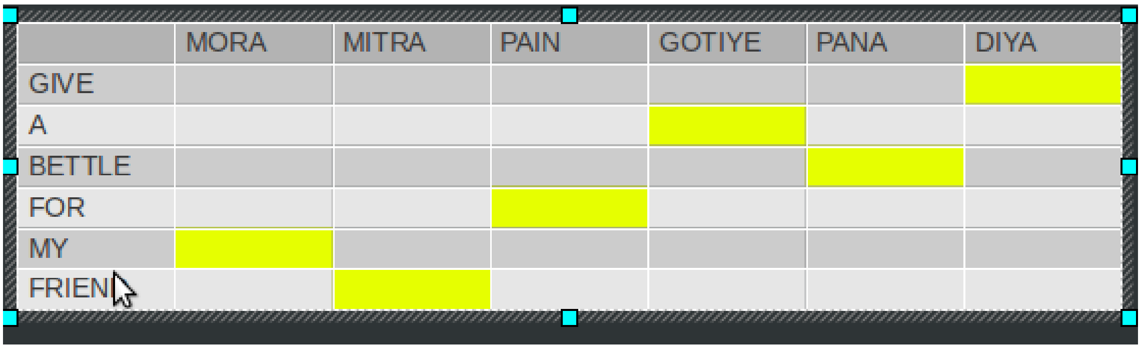
Consider the following example pair from Odia-English bitext.

Odia: mora mitra pain gotiye pan diya
English: give a betel for my friend

Using an aligner, $O \rightarrow E$ alignment and $E \rightarrow O$ alignment are obtained, depicted as below. Taking a union of both alignments, an alignment matrix is obtained as shown below.

mora	my
mitra	friend
pain	for
gotiye	a
pana	betel
diya	give

Table 1.7.1: Odia to English Alignment



	MORA	MITRA	PAIN	GOTIYE	PANA	DIYA
GIVE						🟨
A				🟨		
BETTLE					🟨	
FOR			🟨			
MY	🟨					
FRIEN		🟨				

Figure 1.4: Alignment matrix

1.7.2 Illustration of phrase alignment algorithm using heuristic

To obtain a phrase table, rules are used as stated below.

Rule 1. Given a word-aligned sentence pair
$\langle \mathrm{f}, \mathrm{e}, \sim \rangle$, a rule $\langle f_i^j, e_{i'}^{j'} \rangle$ is an initial phrase pair of $\langle f, e, \sim \rangle$ if and only if:

$$\mathrm{f}_k \sim e_{k'}\ \exists k \in [i,j]\ and\ k' \in [i',j']\ ; (1.7.1)$$
$$\mathrm{f}_k \neq e_{k'}\ \forall k \in [i,j]\ and\ k' \notin [i',j']\ ; (1.7.2)$$
$$\mathrm{f}_k \neq e_{k'}\ \forall k \notin [i,j]\ and\ k' \in [i',j']\ ; (1.7.3)$$

The intuition behind this rule is that phrase f_i^j is translation of phrase $\mathrm{e}_{i'}^{j'}$ if and only if there is some word in French sentence f at index k that is aligned to some word in English sentence at index k′. The second and third rule emphasizes that there is no word in f that is aligned to any word outside phrase e and there is no word in e that is aligned to any word outside phrase f.
Considering our previous example:

X → mora, my
X → mitra, friend
X → mora mitra, my friend
X → gotiye, a
X → pana, betel
X → diya, give
X → gotiye pana diya, give a betel

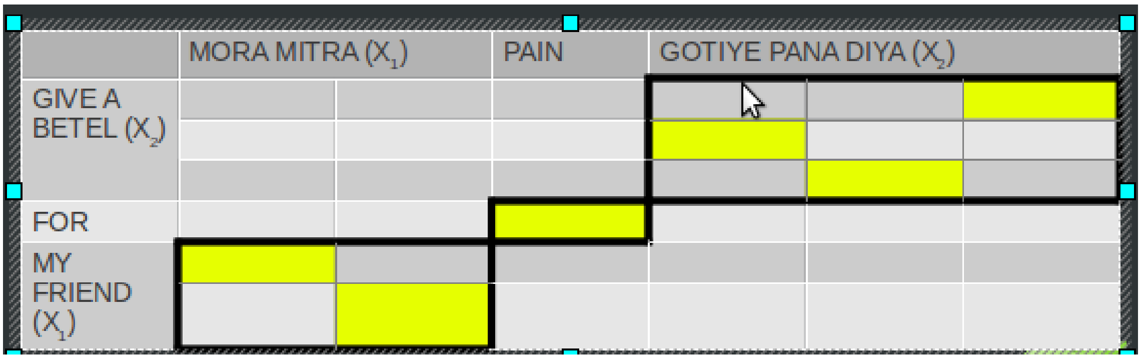
Other phrases can be made as well, but for the sake of translation, they are ignored. Returning to synchronous CFG, more complex rules need to be constructed that has sub-phrases (X) in them.

Rule 2. The rule is as follows:-
$\langle j, e_{i'}^{j'} \rangle$ is an initial phrase pair st $\gamma = \gamma_1 f_i^j \gamma_2$ and $\alpha = \alpha_1 e_{i'}^{j'} \alpha_2$ then X → $\langle \gamma_1 X_k \gamma_2, \alpha_1 X_k \alpha_2 \rangle$ is a rule, where K is an index not used in r.
Going back to our example,

Let r = X → ⟨mora mitra pain gotiye pan diya, give a betel for my friend⟩

If X → ⟨pain gotiye pan, a betel for⟩ is an initial phrase pair such that γ = γ_1 f_i^j γ_2, where γ_1 = mora mitra and γ_2 = diya and $\alpha = \alpha_1 e_{i'}^{j'} \alpha_2$ where α_1 = my friend and α_2 = give, then

X → ⟨ mora mitra X_1 diya, give X_1 my friend⟩



	MORA MITRA (X_1)		PAIN	GOTIYE PANA DIYA (X_2)		
GIVE A BETEL (X_2)						🟨
				🟨		
					🟨	
FOR			🟨			
MY FRIEND (X_1)	🟨					
		🟨				

Figure 1.5: Phrase table

Note: The regions surrounded by black border indicates phrases and their phrase alignments.

1.7.3 Demerits of rule based phrase alignment and solutions to their problems

Notice that the algorithm forms general rules from specific rules. But such an algorithm could lead to unnecessary rules. Consider following example:

$X\rightarrow$ *mora mitra pain, for my friend*
$X\rightarrow$ *gotiye pana diya, give a betel*
$X\rightarrow$ *mora mitra pain gotiye pan diya, give a betel for my friend*
$X\rightarrow X_1\ X_2,\ X_2\ X_1$

It is prohibited for nonterminals to be adjacent on the French side, a major cause of spurious ambiguity. Initial phrases are limited to a length of 10 words on either side. Rules can have at-most two nonterminals. Too many short phrases are not encouraged. A rule must have at-least one pair of aligned words.

1.7.4 Glue Rules

Glue rules facilitate the concatenation of two trees originating form the same nonterminal. Here are the two glue rules. $S\rightarrow S_1\ X_2,\ S_1\ X_2$
$S\rightarrow X_1,\ X_1$
These two rules in conjunction can be used to concatenate discontigous phrases.

1.7.5 Intuition behind using a SCFG

In the first step, we can extract CFG rules for source side language (Odia) from the SCFG rules, and parse the source side sentence with the CFG rules obtained. Let the transfer rules of a SCFG be:-
X→ diya, give
X→ gotiye pana diya, give a betel

Odia CFG
X→ diya
X→ gotiye pana diya

Given an Odia sentence we can obtain a parse tree. Let us go through a Odia to English translation and see what are the stages through which a sentence has to travel to reach the destination. Lets say a user gives our system a test sentence in Odia and is expecting an English sentence as given below.

Odia :-'Bhaina mora mitra pain gotiye pan diya.'
English-'Brother give a betel for my friend.'

1.8 Testing on Odia to English translation

So, input to the system is a sentence in Odia, and a set of SCFG rules extracted from training set. First the decoder filters only the relevant rules from the entire set of grammar rules as shown below.

SCFG for Odia to English translation
$S \rightarrow S_1\ X_2,\ S_1\ X_2$
$S \rightarrow X_1,\ X_1$
$X \rightarrow$ *Bhaina, brother*
$X \rightarrow X_1$ *pain* $X_2.\ X_2$ *for* X_1
$X \rightarrow$ *mora mitra, my friend*
$X \rightarrow$ *gotiye pana diya, give a betel*

These SCFG rules are converted to CFG rules for Odia language only. This is done by taking the source side rules because they are required to parse the given Odia sentence. **Corresponding CFG in Odia**

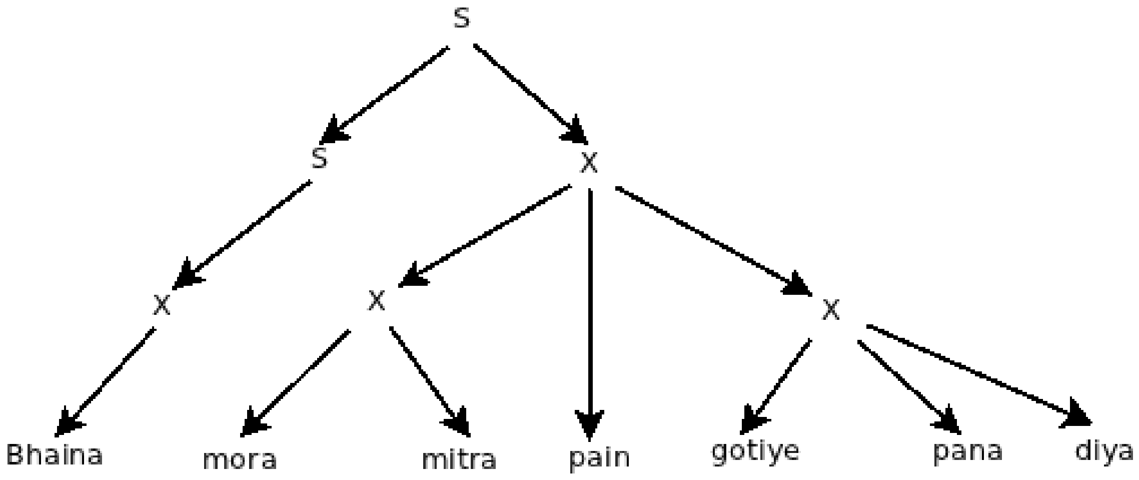


Figure 1.6: Parse Tree in Odia

$S \rightarrow S_1\ X_2$
$S \rightarrow X$
$X \rightarrow Bhaina$
$X \rightarrow X_1\ pain\ X_2$
$X \rightarrow mora\ mitra$
$X \rightarrow gotiye\ pana\ diya$

Step 1:- Parse tree in Odia
Using a CKY parser, the tree in Figure 1.6 is obtained.

Step 2:- Apply transfer rules
We use the transfer rules one by one as shown below to map the Odia parse tree to an English parse tree as shown in Figure 1.7, 1.8, 1.9 and 1.10

$$\mathrm{X} \rightarrow \text{Bhaina, brother} \quad (1)$$
$$\mathrm{X} \rightarrow \mathrm{X}_1 \text{ pain } \mathrm{X}_2.\ \mathrm{X}_2 \text{ for } \mathrm{X}_1 \quad (2)$$
$$\mathrm{X} \rightarrow \text{mora mitra, my friend} \quad (3)$$
$$\mathrm{X} \rightarrow \text{gotiye pana diya, give a betel} \quad (4)$$

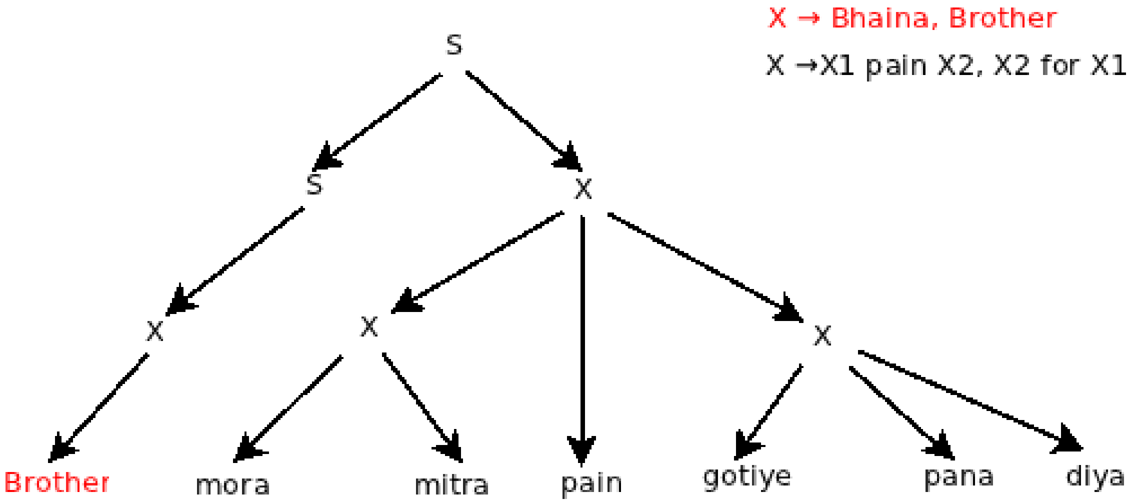


Figure 1.7: The right top corner shows one rule in red which has been applied while the second rule in white is next to be applied to the parse tree. The text mentioned in red implies that text has been translated to English while the text in white indicates that this text is yet to be translated.

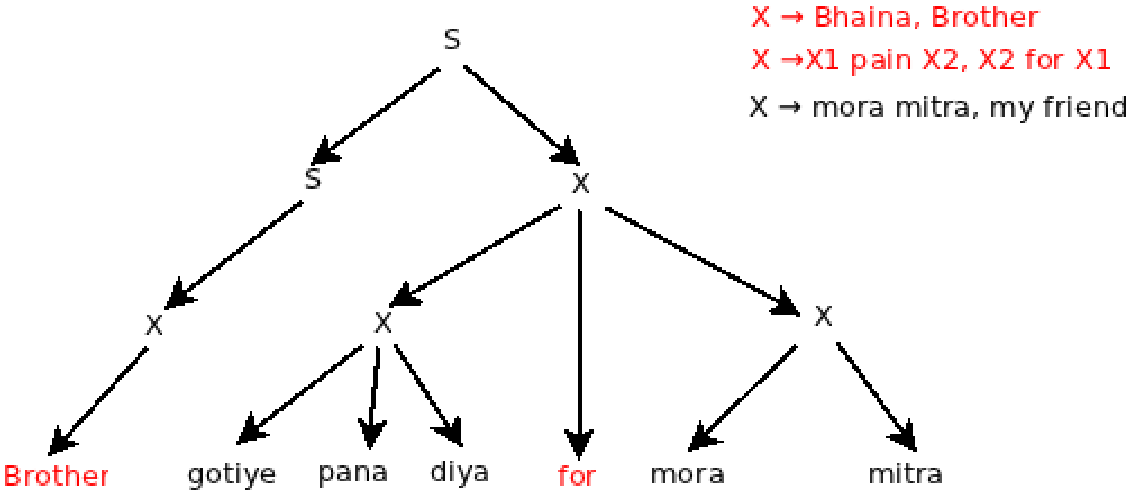


Figure 1.8: This rule replaces terminal pain by for and rotates subtree X_2 and X_1 about terminal for thus accounting for local reordering at phrase level.

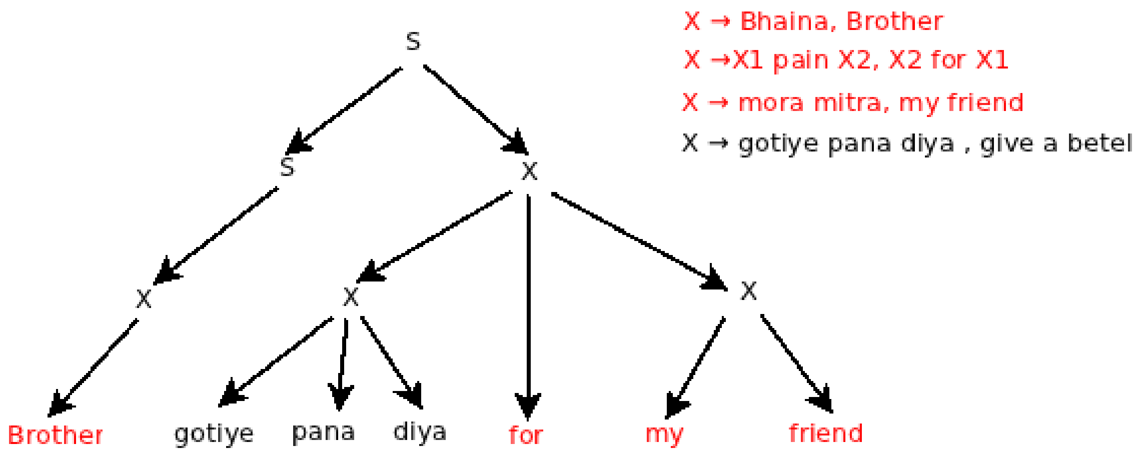


Figure 1.9: Parse Tree after applying rule #3.

Step 5:- Apply rule 4

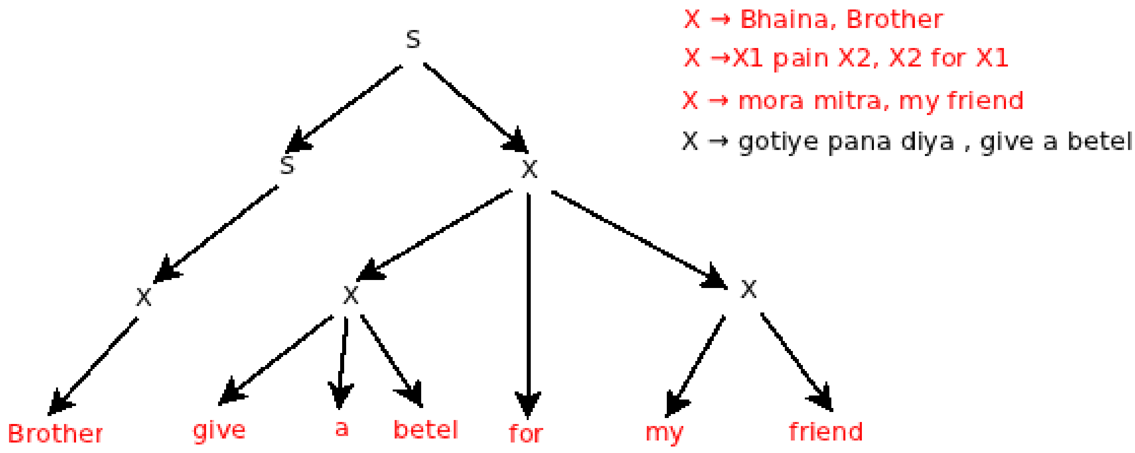


Figure 1.10: Parse Tree after applying rule #4.

Output
English:- "Brother give a betel for my friend."

1.9 Open source hierarchical phrase based machine translation system

Large-scale parsing-based statistical machine translation (e.g. Chiang [2007], Quirk et al. [2005], Galley et al. [2006], Liu et al. [2006]) has made remarkable progress in the last few years. However most of the systems mentioned above are not open source and hence are not easily available for research. This results in a high barrier for new researcher to understand previous systems and improve them. In this scenario, open source can play a huge role in improving the number of experiments and magnitude of research going on in MT world. In the following topics, we present two of the well known open source hierarchical phrase-based MT systems.

1.10 JOSHUA

Joshua is an open source statistical MT toolkit. Joshua implements all of the algorithms required for synchronous CFGs: chart parsing, n gram language model integration, beam and cube pruning, and k-best extraction. The toolkit also includes a module for suffix array grammar extraction and minimum error rate training (MERT). To accommodate scalability, it uses parallel and distributed computing techniques. It has been demonstrated that the toolkit achieved state-of-the-art translation performance on the WMT09 French-English translation task.

1.10.1 Main functionalities

In this part, we have discussed the various functionalities of Joshua pipeline.

Training corpus sub sampling

Instead of using the entire corpus for extracting grammar, only a sample of the corpus is used as proposed by Kishore Papineni. This method works as follows: for the sentences in the development and test set that are to be translated, every n gram up to length of 10 is gathered in a map W. Only those sentence pairs are selected from the training set that contains any n-gram found in W with a count of less than k. Every sentence that is selected causes an increment of the n-grams in W present in it by their count in that sentence. The reason is that similar sentences, *i.e.*, sentences containing the same n- grams will be rejected subsequently. This helps in reducing redundancy in new training set and less time taken while training.

Suffix Array Grammar Extraction

Hierarchical phrase-based MT requires grammar extracted from parallel corpus but in real translation tasks, grammar are too big and often violate memory constraints. In such tasks,feature calculation is damn expensive considering the time required; huge sets of extracted rules must be sorted in opposite direction to obtain features like translation probability p (f | e)and p (e | f) (Koehn et al. [2003]). In case the training data is changed, the extraction steps have to be re run. To alleviate such issues, a source language suffix array is used to extract only those rules that will be useful in translation following Callison-Burch et al. [2005]. This reduces the rule set compared to techniques that use the entire training set from extracting rules.

Decoding Algorithms

In this part, we describe the various sub-functionalities of the decoding algorithms as described in Li et al. [2010].

Grammar Formalism The decoder implements a synchronous context free grammar (SCFG) of the kind described by Heiro. (Chiang [2005]).

Chart Parsing Given a source sentence, the decoder produces 1-best and k-best translation using a CKY parser. The decoding algorithm maintains a chart, which contains an array of cells. Each cell in turn maintains a list of proven items. The parsing process starts with axioms, and proceeds by applying the inference rules repeatedly to prove new items until proving a goal item. Whenever the parser proves a new item, it adds the item to the appropriate chart cell. The item also maintains back pointer to antecedent items, which are used for k-best extraction.

Pruning Severe pruning is required to make decoding tractable. The decoder incorporates beam pruning and cube pruning (Chiang [2005]).

Hypergraph and k-best extraction For each source language sentence, the chart parsing algorithm produces a hypergraph, that contains an exponential set of likely derivation hypotheses. Using k-best algorithm, the decoder extracts the top k translations for each sentence.

Parallel and Distributed decoding They also work on parallel decoding and distributed language model using multi core and multi processor architecture and distributed computing techniques.

1.10.2 Language Model

They implement an ngram language model using a n-gram scoring function in Java. This java implementation can read ARPA fromat provided by SRILM toolkit and hence the decoder can be used independently from SRILM. They also developed their own code that allows the decoder to use the SRILM toolkit to read and score n-grams.

1.10.3 MERT

JOSHUA's MERT module is called ZMERT as described earlier. It provides a simple java implementation to efficiently determine weights for the log-linear model used for scoring translation candidates to maximize performance on a development set as measured by an automatic evaluation metric, such as BLEU.

1.11 Moses

Moses Koehn et al. [2007] is also an open source phrase-based MT system. Recently it has started developing hierarchical phrase-based MT to become a complete toolkit. Moses was developed prior to JOSHUA. Hence it brought in a completely out of the box translation toolkit for academic research. Developed by several scientists in the University of Edinburgh, it gave big boost to MT research. Also it brought new concepts like a pipeline in the era of MT systems wherein you just give a shell command, the pipeline is executed automatically making the system user friendly. The pipeline consists of three different stages training, testing and tuning.

The developers of Moses were concerned about phrase-based model's limitations which translated chucks of words without making any use of linguistic information like morphological, syntactic or semantic. So they integrated factor-based translation in which every word is morphologically analyzed and then translated. This certainly improves the quality of translation.

1.11.1 Factored Translation Model

Non factored SMT deals with chunks of words and has one phrase table as explained in **??**

1.12 Example of phrase based MT lagging

Translate:-
I am buying you a green cat.
“मै आप के लिये एक हरे रन्ग की बिल्ली खरीद रहा हून.
Using phrase dictionary.
I → *मै*
am buying → *खरीद रहा हून*
you → *आप के लिये*
a → *एक*
green cat → *हरे रन्ग की बिल्ली*

In factored translation , the phrases may be augmented with linguistic information like lemma or POS tags.

$$\begin{pmatrix} billi \\ NN \\ billi \\ sing/fem \end{pmatrix} \rightarrow \begin{pmatrix} cat \\ NN \\ cat \\ sing \end{pmatrix} \tag{1.12.1}$$

Mapping of source phrases to target phrases can be done in a number of steps so that different factors can be modelled separately thereby reducing dependecies between models and improving flexibility.

For ex:- sing/pl masc/fem should not depend on POS tag.

घरो → घर + “ओ ”→ Lemma⟨घर ⟩ POS⟨NN⟩ mod⟨pl⟩ $\overset{\text{translate to english}}{=}$ Lemma⟨house ⟩ POS⟨NN⟩ mod⟨pl⟩ → house + “s ”→ houses.

So the surface form was first transformed to lemma and surface forms, then the target was built from the lemma and other linguistic information. This reduces the size of phrase table considerably.

1.12.1 Toolkit

It consists of all the components needed to preprocess data, train the language models and the translation models. For tuning, it uses MERT and BLEU for evaluating the resulting translations. Moses uses GIZA++ for alignment and SRILM for language modeling. The toolkit is available online as open source under sourceforge.

The decoder is the core component of the toolkit which was adopted from Pharaoh to attract the interests of followers of Pharaoh. In order for the toolkit to be adopted by the community, and to make it easy for others to contribute to the project, the following principles were kept in mind:

- Accessibility
- Easy to maintain
- Flexibility
- Easy for distributed team development
- Portability

It was developed in C++ for efficiency and followed modular, object oriented design.

References

Brown, P. F., Cocke, J., Pietra, S. A. D., Pietra, V. J. D., Jelinek, F., Lafferty, J. D., Mercer, R. L., and Roossin, P. S. (1990). A statistical approach to machine translation. *Comput. Linguist.*, 16(2):79–85.

Callison-Burch, C., Bannard, C., and Schroeder, J. (2005). Scaling phrase-based statistical machine translation to larger corpora and longer phrases. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, ACL '05, pages 255–262, Stroudsburg, PA, USA. Association for Computational Linguistics.

Chiang, D. (2005). A hierarchical phrase-based model for statistical machine translation. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, ACL '05, pages 263–270, Stroudsburg, PA, USA. Association for Computational Linguistics.

Chiang, D. (2007). Hierarchical phrase-based translation. *Comput. Linguist.*, 33(2):201–228.

Galley, M., Graehl, J., Knight, K., Marcu, D., DeNeefe, S., Wang, W., and Thayer, I. (2006). Scalable inference and training of context-rich syntactic translation models. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, ACL-44, pages 961–968, Stroudsburg, PA, USA. Association for Computational Linguistics.

Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A., and Herbst, E. (2007). Moses: open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, ACL '07, pages 177–180, Stroudsburg, PA, USA. Association for Computational Linguistics.

Koehn, P., Och, F. J., and Marcu, D. (2003). Statistical phrase-based translation. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, NAACL '03, pages 48–54, Stroudsburg, PA, USA. Association for Computational Linguistics.

Li, Z., Callison-Burch, C., Dyer, C., Ganitkevitch, J., Irvine, A., Khudanpur, S., Schwartz, L., Thornton, W. N. G., Wang, Z., Weese, J., and Zaidan, O. F. (2010). Joshua 2.0: a toolkit for parsing-based machine translation with syntax, semirings, discriminative training and other goodies. In *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*, WMT '10, pages 133–137, Stroudsburg, PA, USA. Association for Computational Linguistics.

Liu, Y., Liu, Q., and Lin, S. (2006). Tree-to-string alignment template for statistical machine translation. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, ACL-44, pages 609–616, Stroudsburg, PA, USA. Association for Computational Linguistics.

Nagao, M. (1984). A framework of a mechanical translation between japanese and english by analogy principle. In *Proc. of the international NATO symposium on Artificial and human intelligence*, pages 173–180, New York, NY, USA. Elsevier North-Holland, Inc.

Quirk, C., Menezes, A., and Cherry, C. (2005). Dependency treelet translation: syntactically informed phrasal smt. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, ACL '05, pages 271–279, Stroudsburg, PA, USA. Association for Computational Linguistics.

Chapter 2

2.1 Phrase based translation of a Hindi sentence to English sentence

भारत का प्रधान मन्त्री {bhaarata kaa pradhaana mantrii} {India of Prime Minister}	→ Prime Minister of India
जापान का प्रधान मन्त्री {jaapaana kaa pradhaana mantrii} {Japan of Prime Minister}	→ Prime Minister of Japan
चीन का प्रधान मन्त्री {ciina kaa pradhaana mantrii} {China of Prime Minister}	→ Finance Minister of China
भारत का राष्ट्रीय पक्षी {bhaarata kaa raastriiya pakshii} {India of National bird}	→ National bird of India

2.2 Example to establish reordering

For example :- (This mapping was observed during training)

भारत का प्रधान मन्त्री → Prime Minister of India

If a similar phrase appear during testing,

भारत का राष्ट्रीय पक्षी

Even if it had the translations of words in the above phrase,

भारत → India
का → of
राष्ट्रीय पक्षी → National bird

This will give an incorrect output like:-

भारत का राष्ट्रीय पक्षी → India of National bird

2.3 Websites for Gazetteer list

- babynames
- Surname
- http://en.wiktionary.org/wiki/Appendix:Indian_surnames_(Arora) (Bunt) (Chitpavan) (Deshastha_Brahmin) (Goan_Christian) (Paravar) (Shivalli)Indian surnames
- http://www.indiacom.com/yellowpage/telephonedirectories.aspTelephone directory

2.4 Examples of noisy data in CoNll corpus

1. HYPERLINK-`http://en.wikipedia.org/wiki/`
2. Bracketed information:- (DoD) {Common Access Card}
3. Citations:- (Ben, 2008)
4. Presence of sentence pairs without any changes.

   Ex:-*Our current population is 6 billion people and it is still growing exponentially → Our current population is 6 billion people and it is still growing exponentially.*

2.5 Grammar correction example

Input to Hi-En translation system is:-	*सेन्ट्रल लंडन में गिरा प्लेन*
Expected output is:-	*plane down in central london.*
Output from Hi-En translation system is:-	*central down in london plane.*
Input to Grammar correction is:-	*central down in london plane.*
The output is:-	*plane down in central london.*

2.6 Single reference translation

System A: *Israeli officials responsibility of airport safety*
Reference: *Israeli officials are responsible for airport security*
SYSTEM B: *airport security Israeli officials are responsible*

2.7 Multiple Reference Translations

System:-
Israeli officials responsibility of airport safety

References:-

Israeli officials are responsible for airport security
Israel is in charge of the security at this airport
The security work for this airport is the responsibility of the Israel government
Israeli side was in charge of the security of this airport

2.8 Translation models

We experimented with the following setup:-

- Phrase Based Translation
  - Moses trained on Indian Parallel Corpora
  - Moses trained on Gyan Nidhi Corpus
- Hierarchical Phrase Based Translation
  - Joshua trained on Indian Parallel Corpora
  - Joshua trained on Gyan Nidhi Corpus
- Factor Based Translation Model
  - Moses Trained on Factored Gyan Nidhi Corpus

2.8.1 Factor-Based Translation Model

The setup used for factor based translation model is as follows:-

- Factors used:
  - word + POS tags + stem
- Tools used:
  - Hindi Side:
    - * Hindi POS tagger from CFILT, IIT Bombay
    - * Hindi Rule-Based Stemmer from Lucene
  - English Side:
    - * Stanford POS Tagger
    - * English Rule-Based Stemmer from Lucene

The configuration for factors has been done as follows:-

- Input Factors (Hindi side): word — POS — stem
- Output Factors (English side): word — POS
- Translation Factors:
  - Stem - stem
  - Stem - word, POS
  - Word - word, POS
- Generation Factors:
  - Stem POS
  - POS, stem word

2.9 Hindi-english translation

Hindi - *अंतरिक्ष में प्रथम भारतीय अप्रैल १९८४ में भारत ने अंतरिक्ष विज्ञान के क्षेत्र में एक और सफ़लता प्राप्त की जब पेहला भारतीय अंतरिक्ष यात्री राकेश शर्मा जो भारतीय वायूसेना के एक पाईलट थे अंतरिक्ष पहुँचे*

वैधानिक प्रतिरोध

Transliteration - *Antariksha mein pratham bharatiya: aprail 1984 mein bharat ne antariksha vigyan ke kshetra mein ek aur safalta prapta kee jab pehla bharatiya antariksha yatri Rakesh Sharma jo bharatiya vayusena ke ek paylat the antariksha pahunche.*

Gloss - *Space in first Indian: April 1984 in India had space science in field in one more success got did when first Indian space traveller Rakesh Sharma who Indian airforce of a pilot was space reached.*

English - *first indian: april in space in 1984 india had a in space science and got success when the first indian space travelers rakesh sharma which was a* पाईलट *of indian* वायूसेना *operation but the space.*