The Reordering Problem in Statistical Machine Translation

Ph.D Seminar Report

Submitted in partial fulfillment of the requirements
for the degree of

Doctorate of Philosophy

by

Anoop Kunchukuttan
Roll No: 114056004

under the guidance of

Prof. Pushpak Bhattacharyya

Department of Computer Science and Engineering
Indian Institute of Technology, Bombay
Mumbai

Declaration

I declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

(Signature)

(Name of Student)

(Roll No)

Date:

Acknowledgments

I would like to thank Prof. Pushpak Bhattacharyya for his keen guidance and constant support. His enthusiasm has always kept me in high spirits and I hope some of that rubs off on me. I also thank all my colleagues in the SMT group who have participated enthusiastically in the weekly meetings. The stimulating thoughts at these meetings have no doubt been good food for thought. Finally, thanks to Mitesh, Balamurali, Brijesh, Samir, Abhijit and Rajen for a very vibrant environment in TCS lab that I have completely enjoyed.

Anoop Kunchukuttan

Abstract

Reordering of words is one of the most visible changes when translating a sentence from one language to another. The reordering problem has always been a central concern for machine translation. In this report, we look at the reordering problem in the context of Statistical Machine Translation. We first study and classify the reordering divergences between languages. We emphasize the contribution that linguistic understanding of these divergence patterns plays in developing efficient and effective solutions to the reordering problem. We study how the reordering is modeled in the major SMT models and analyze these models with respect to coverage of divergence patterns, model complexity and use of linguistic resources. Machine translation is distinguished from other machine learning problems with respect to the complexity of inference, known as decoding in machine translation terminology. We describe methods used to efficiently perform decoding while attempting to get the best possible reordering. We look at extensions beyond the noisy channel model, in the form of source reordering and discriminative re-ranking, which attempt to further improve reordering. In this report, our endeavour has been to do a holistic study of all aspects of the reordering problem, across all components of an SMT system and identify potential areas of research.

Keywords: statistical machine translation, reordering, decoding

Contents

1 Introduction **1**

1.1 Overview 1

1.2 Machine Translation Paradigms 2

1.3 Basics of Statistical Machine Translation 3

1.3.1 The principle 3

1.3.2 Noisy Channel Model 3

1.4 Organization of the report 4

2 Reordering Divergences **6**

2.1 Introduction 6

2.2 Intra-Clausal divergences 6

2.2.1 Constituent Order Divergence 7

2.2.2 Head-Modifier Location Divergence 8

2.3 Inter-Clausal divergences 9

3 Modeling reordering **11**

3.1 Word based Models 11

3.1.1 Alignment based process 12

3.1.2 HMM based model 13

3.1.3 Fertility based models 14

3.2 Phrase based models 16

3.2.1 Learning Phrase based models 17

3.2.2 Reordering Phrase based models 17
3.3 Syntax based Models 19

4 Handling reordering during decoding 22
4.1 Search space for a decoder 22
4.2 Stack based decoding 23
4.3 Reordering Constraints 24
4.3.1 IBM constraint 24
4.3.2 ITG constraint 25
4.3.3 Clause boundary constraint 25
4.4 Enumerating good hypotheses 26

5 Pre-processing and post-processing to improve reordering 27
5.1 Source side reordering 27
5.1.1 Need for source side reordering 27
5.1.2 How does source side reordering help? 28
5.1.3 Reordering with hand coded rules 29
5.1.4 Reordering with automatically extracted rules 30
5.2 Re-ranking of translation hypothesis 32
5.2.1 A discriminative approach to translation 32
5.2.2 Why re-rank candidate translations? 32
5.2.3 Features for re-ranking 33

6 Conclusion and Future work 34
6.1 Summary and Observations 34
6.2 Future Directions 35

List of Figures

1.1 Noisy Channel Model 4

3.1 Alignment of two sentences 12
3.2 Alignment matrix to learn phrase table 18
3.3 Syntax based SMT system [Src: [YK01]] 20
3.4 Reordering parameters in syntax-based SMT [Src: [YK01]] . . 21

4.1 IBM constraints [Src: [ZN03]] 25
4.2 ITG constraints [Src: [ZN03]] 26

5.1 Learning Reordering patterns using chunks and word alignments 31

Chapter 1

Introduction

1.1 Overview

Machine translation has often been considered one of the most challenging Natural Language Processing tasks. A lot of research has gone into Machine Translation since the advent of computing, but the problem is far from solved.

> *"After only forty years of research and development in MT, I feel about its condition a little as Mao Tse-Tung is said to have felt about the significance of the French Revolution: that it was too early to tell."*
> - Yorick Wilks

While translating a sentence from one language to another, various consistent patterns of divergence between the languages are observed. These could be at the morphological, syntactic or syntactico-semantic levels. For instance, languages may differ in the morphological complexity associated with words, or have differences in how lexico-semantic roles are represented, etc. Machine translation systems have to account for these divergences. One of the most important divergences is the *reordering divergence*. This divergence represents the difference in the order in which various linguistic components are linearized in different languages.

S	V	O
Amit	met	the advocate
अमित	वकील से	मिला
S	O	V
amita	vakiila se	milaa

In the example shown here, it can be seen that there is a reordering of the corresponding components of the sentence pair. The verb follows the subject in English, whereas it moves to the end of the sentence in Hindi.

Translation systems need to account for reordering divergence. In this report, we analyze the reordering divergences that typically occur. Very broadly, machine translation systems fall into two categories: *those which rely on symbolic processing methods and those built on statistical methods.* We review the methods used for handling the reordering in the context of Statistical Machine Translation (SMT)systems. We analyze these solutions with respect to their ability to handle the reordering divergences discussed in the report. We try to identify the major research problems that need attention for tackling the reordering problem.

1.2 Machine Translation Paradigms

In the classic symbolic processing/rule-based methods, there has been an emphasis on understanding the linguistic structure of the languages involved, developing knowledge representation schemes needed to capture these structures and building programs to transform this rich representation of language from one language to another. The key to the success of such systems has been the availability of high quality expert linguistic knowledge framework.

Handling the reordering problem in a rule-based system involves writing extensive transfer rules to transform syntactic structures from source language to the target language. In an inter-lingua based, the challenge is even greater, where the aim is to encode the input language into a language independent knowledge representation. The main arguments against symbolic/rule-based machine translation systems are that they are not scalable to new language pairs - both in terms of the availability of linguistic expertise, and the cost involved.

In recent years, with the availability of large corpora and falling costs in computing power, statistical methods have been widely investigated for MT as in other areas of NLP. The statistical argument is to model the translation problem to the extent required to learn translation patterns from the corpus, rather than invest time in building rules. While it is implausible to believe that machine translation systems can be built without any linguistic knowledge, the emphasis is to minimize the human effort involved in creating linguistic knowledge and resources by learning these from the data.

Many SMT systems have been proposed which represent the translation problem at various levels of complexity. For the reordering problem too, many methods have been proposed having different levels of complexity and handling different kinds of reorderings. The endeavour of this report is to review the research work done in all components of an SMT system for tackling the reordering problem.

1.3 Basics of Statistical Machine Translation

1.3.1 The principle

In Statistical Machine Translation (SMT), translation is modeled as a stochastic process. The stochastic model is characterized by some parameters, and the goal is to learn these parameters from the training corpus. The training corpus, called a *parallel corpus*, contains a large number of pairs of translated sentences in the source and target languages. Using well known parameter estimation principles from machine learning like maximum likelihood and maximum entropy, the SMT method provides a principled way of handling uncertainty in the translation process and achieving generalization. This is one of the strengths of the SMT method: the ability to handle uncertainty and ambiguity using corpus evidence.

However, it should be mentioned that the quality of the machine translation system depends on the model bias i.e. the linguistic process which is assumed to generate the translations - for learning cannot happen in a bias. Thus, the statistical method, as it stands today, does not make a claim of using no linguistic resources. Rather the attempt is to find the right balance in automatic knowledge acquisition and investing human expertise. Of course, it becomes more difficult to model the translation process in a stochastic framework with increasing linguistic knowledge.

1.3.2 Noisy Channel Model

> *"When I look at an article in Russian, I say: This is really written in English, but it has been coded in some strange symbols. I will now proceed to decode."*
> - Warren Weaver.

The noisy channel model is the stochastic process the explains the translation process [BPPM93]. It likens the translation process to a cipher-decoding task. The noisy channel defines a generative process where a sentence e in language E is transformed to a sentence f in language F. The task in machine translation is to decode the message f to retrieve the message e. This is equivalent to translating the sentence in language E to a sentence in language F. This can be written as:

$$e^* = \arg\max_e P(e|f)$$

Using Bayes' Rule this can be written as,

$$e^* = \arg\max_e P(f|e)P(e)$$

$P(f|e)$ is called the *translation model* which plays the role of explaining the translation of a source sentence into the target sentence. $P(e)$ is called the

Figure 1.1: Noisy Channel Model

language model, which models how good (grammatical) the target language sentence is. The translation model thus captures the *fidelity* of the translation system, whereas the language model captures the *fluency* of the target language sentence.

The first task in building an SMT system is to **model** the translation and language model. The reordering model will be part of the translation model. The next task is to **learn** the parameters of the models using the corpus. The translation model is typically learnt from a parallel corpus, whereas the language model can be learnt from a larger monolingual corpus in the target language. Finally, when a new source language sentence is made available for translation, it has to be **decoded**. Decoding is the process of searching for a target translation which is most likely given the source sentence, as per the language model and the translation model. The decoder is responsible for generating the words of the translated sentence in the correct order.

1.4 Organization of the report

In Chapter 2 we describe the various reordering patterns between languages that need to be tackled by machine translation systems. Chapter 3 studies various SMT models, and analyzes them from the point of modeling reordering. It analyzes the abilities of various SMT models to model the reordering divergences described in Chapter 2. Decoding is a very challenging problem for SMT, because of the large output space - partly because of reordering of the words. Chapter 4 studies search techniques for decoding in SMT. It studies the use of reordering constraints to limit the search space, and their effect on the quality of translation. We also discuss enhancing the translation search space by instructing the decoder to include specified hypotheses. The traditional noisy channel model alone is not sufficient to solve the reordering problem. Therefore, in Chapter 5, we discuss preprocessing steps on the source side to order the source language words to correspond to the target word language order. We also discuss re-ranking of translation can-

didates using discriminative modeling as a post-processing step using many linguistically motivated features. Finally, Chapter 6 concludes the report with a summary of the survey and identifying potential directions of work.

Chapter 2

Reordering Divergences

2.1 Introduction

Translation of a sentence from one language to another often results in a sentence whose form is very different from that of the source language. There are distinct patterns in these cross linguistic differences, which are referred to as *translation divergences*. Translation divergences have been widely studied with the aim of understanding common patterns of divergences observed across languages, which can aid systematic and modular design of machine translation systems [Dor94, DPB03, ST05].

Reordering of words and other linguistic constituents is one of the most important divergences observed in language translation. In this chapter, we present a classification of reordering divergences that are observed in language translation. We describe different reordering patterns observed in translation. We use English-Hindi language pair to illustrate reordering divergences. However, some of these patterns are of a general nature or emphasize an important reordering consideration that applies to translation across different language pairs. An understanding of these reordering divergences is essential to clearly define the reordering issues to be handled in machine translation. Machine translation systems can exploit knowledge of these reordering patterns for building effective computational systems.

2.2 Intra-Clausal divergences

Many reorderings occur within corresponding clauses of translated sentences, and constitute the fundamental reorderings occurring in a translation. Thus clauses impose some constraints on the possible reorderings. This knowledge can used to make the search for correct reorderings tractable.

This section describes many of the common reordering divergence patterns seen within a clause translation.

2.2.1 Constituent Order Divergence

Constituent word order refers to the relative positions to the subject (S), object(O) and verb(V) in a clause. There are six possible constituent orderings:

SVO, SOV, VSO, VOS, OSV, OVS

Languages can be classified on the basis of constituent order. Hindi, all Indo-Aryan languages and Dravidian languages are SOV languages, while English, Chinese, Russian are SVO languages. In fact, these two groups account for 87% of the world's unique languages [1].

The fundamental constituent order divergence between English and Hindi [DPB03] can be expressed as:

SVO $\leftrightarrow$ SOV

This is illustrated with the example below:

S	V	O
Amit	met	the advocate
अमित	वकील से	मिला
S	O	V
amita	vakiila se	milaa

Languages are, however, not rigid with respect to the constituent order. Hindi, for example, is a relatively free word order language with SOV being the dominant word order. The Hindi sentence illustrated above could also have been written as:

वकील से अमित मिला
vakiila se amita milaa

English, on the other hand, is particular about word order and changing the word order may result in changing the semantics. The reordering shown below does not mean the same as the original English sentence.

The advocate met Amit.

Translation involving a free word order source language can be a challenge for SMT system, since they rely corpus support will be divided among different possible constituent orders.

We can extend this analysis of constituent order divergence to include prepositional phrases (P) modifying the verb. As an illustration,

S	V	O	P
Amit	met	the advocate	in the evening
अमित	शाम को	वकील से	मिला
S	P	O	V
amita	"saama ko	vakiila se	milaa

[1] http://en.wikipedia.org/wiki/Word_order#Constituent_word_orders

This divergence can be captured by the following rule:

SVOP $\leftrightarrow$ SPOV

Fortunately, the same rules applies even if the prepositional phrase were attached to the noun. Thus, PP attachment does not cause a change in the reordering rule for English-Hindi. However, this may not be the case for all language pairs and hence attachment ambiguity must be considered for reordering.

If the presence of an object complement is considered, we have the following rule,

SVOO$_c$P $\leftrightarrow$ SPOO$_c$V

illustrated by,

S	V	O	O$_c$	P
I	made	him	the President	during the meeting
मैने	बैठक मे	उसे	अध्यक्श	बनाया
S	P	O	O$_c$	V
maine	bai.Taka me	use	adhyak"sa	banaayaa

2.2.2 Head-Modifier Location Divergence

In addition, each of the clause's constituents may undergo internal reordering too. Reordering in the relative positions of heads and their modifiers is an example of this category. These reorderings are typically local in nature. Various types of head-modifier reorderings seen in the English-Hindi language pair are:

Genitive Case: The head and modifier noun phrases switch positions in the case of a noun phrase involving a genitive case.

> The King of Nepal
> नेपाल के राजा
> nepaala ke raajaa

Adverb-Verb Relative Position: The adverb which generally follows the verb in English, precedes it in Hindi.

> He runs fast
> वह तेज़ भागता है
> vaha teza bhaagataa hai

Aspect, Modality modifiers relative to the verb: In English, the modal and some aspect modifiers precede the verb whereas in Hindi these modifiers follow the verb.

> You should meet him
> आपको उनसे मिलना चाहिये
> aapako unase milanaa caahiye

Negation particle relative to verb In English, the negation modifier precedes the verb, whereas in Hindi it follows the verb.

I did not call you.
मैने तुम्हे नही बुलाया
maine tumhe nahii bulaayaa

2.3 Inter-Clausal divergences

The clause forms an important unit for understanding reordering. Clauses are either independent or dependent. Depending on the number and independence of clauses, sentences are either simple, compound or complex. In all these sentence types, we can observe the following reordering interactions between clauses in the sentence pair:

- Generally, word translations do not cross clause boundaries.
- Generally, the relative positions of the clauses are retained. But the order of clauses is pretty free order as long as the connecting conjunctions are properly placed.
- Within each clause, the intra-clausal rules mentioned in the previous section hold without any interference from other clauses.

Thus, clauses define a partition of words in a sentence, that restrict the possible reorderings. Since clauses can be embedded in one another, a recursive partition of the words is defined. In a machine translation system, this knowledge can serve as an important constraint to do efficient and accurate translation. Hence, detection of clause boundaries is an important component for obtaining correct reordering in translations.

Following are examples of the above mentioned interactions for compound and complex sentences:

- **Compound Sentence**
  A visitor came to my home, so I could not go to the market.
  घर पर एक मेहमान आये थे, इसलिये मै बाज़ार नही जा पाया
  Gara para eka mehamaana aaye the, isaliye mai baazaara nahii jaa paayaa
- **Complex Sentence**
  I left early yesterday because I had to go to a meeting.
  मै कल जल्दी चला गया क्योंकि मुझे एक बैठक मे जाना था
  mai kala jaldii calaa gayaa kyoMki mujhe eka bai.Taka me jaanaa thaa

But there are a few cases where a variation from this canonical behaviour is observed, as illustrated below:

- Infinitive clause: The infinitive clause, acting as a noun phrase, is reordered as per the constituent order divergence between English-Hindi, thus being embedded into the main clause [Sin03].

  I asked him to buy a ticket
  मैने उसे टिकेट खरीदने को कहा
  maine use .tike.ta Kariidane ko kahaa

- Infixing of dependent adjective clause: The main clause is infixed into the dependent adjective clause as shown below:

  The decision we took was correct
  हमने जो निर्णय लिया वह सही था
  hamane jo nir.naya liyaa vaha sahii thaa

- *vaalaa* participle in Hindi: Some relative clauses in English can be instantiated by the *vaalaa* participle in Hindi, in which case the modifier precedes the head in the Hindi translation [ST05].

  The boy who sells fruits did not come today
  फल बेचने वाला लडका आज नही आया है
  Pala becane vaalaa la.dakaa aaja nahii aayaa hai

- Long distance dependencies: In a Wh-question in English, the wh-pronoun is displaced from its expected location as object of the verb to the start of the sentence. This displacement can occur across many phrase and clause boundaries as shown in the example. However, such a displacement does not happen in Hindi, leading to a reordering divergence, which is very difficult to capture through syntactic rules. The reordering spans many clause and phrase boundaries.

  What did you say you wished to buy _?
  तुम क्या खरीदना चाहते हो यह कहा?
  tuma kyaa khariidanaa caahate ho yaha kahaa?

Chapter 3

Modeling reordering

As we saw in Chapter 1, statistical modeling of the translation model forms an important component in Statistical Machine Translation. Reordering is a central concern in statistical modeling for which many approaches have been proposed. These approaches vary in the linguistic knowledge they require, the ability to model different reordering divergences and the computational complexity in learning and decoding.

These approaches broadly fall into three categories:

1. Word based models
2. Phrase based models
3. Syntax based models

In this chapter, we discuss these approaches and their variants.

3.1 Word based Models

Word based models model the sentence translation problem as a *string-to-string* translation problem. Individual words are translated without taking into account the target or source language syntax. In the absence of syntactic knowledge, the mapping of words in the source language to words in the target language is represented by a structure called the **alignment**. Figure 3.1 shows a sentence pair and their alignment.

> An alignment ($\mathbf{a}$) of a sentence pair ($\mathbf{f}, \mathbf{e}$) in a source language (F) and target language (E) is defined as a relation between the positions of words in the two sentences [ON03] i.e.
>
> $$\mathbf{a} \in \mathcal{A} \subseteq \{(j, i) : j = 1, \ldots, len(\mathbf{e}); i = 1 \cdots len(\mathbf{f})\}$$

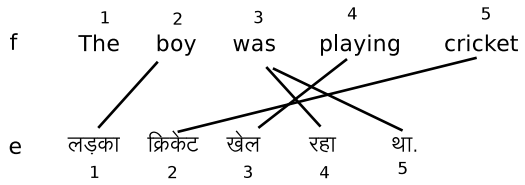

Figure 3.1: Alignment of two sentences

While the sentence pair is available in training data, the alignments are not known. The same translation could be generated by different alignments. Hence, the translation model is obtained by marginalizing over all possible alignments:

$$Pr(\mathbf{f}|\mathbf{e}) = \sum_{\mathbf{a}} Pr(\mathbf{f}, \mathbf{a}|\mathbf{e})$$

The reordering problem in word based models is to learn this alignment relation. The popular IBM word models and its variants propose three generative processes to explain the sentence translation process and word reordering. In these models, the alignment is simply represented as a function, a_j, (as opposed to the relation $\mathcal{A}$) from word position in the source language (F) to target language(E).

3.1.1 Alignment based process

IBM Models 1 and 2 are explained by the following generative translation process. Given sentence $\mathbf{e}$ of length l, the sentence $\mathbf{f}$ with alignment $\mathbf{a}$ is generated as,

- Select the length (m) of the sentence $\mathbf{f}$
- For each position j in $\mathbf{f}$:
  - Choose the aligned position a_j in sentence e, depending on the words chosen and positions aligned so far.
  - Choose the word f_j at position j.

This leads to the following translation model,

$$Pr(\mathbf{f}, \mathbf{a}|\mathbf{e}) = Pr(m|\mathbf{e}) \prod_{j-1}^{m} Pr(a_j|a_1^{j-1}, f_1^{j-1}, m, \mathbf{e}) Pr(f_j|a_1^j, f_1^{j-1}, m, \mathbf{e}) \tag{3.1}$$

While Equation 3.1 specifies the probability distribution completely, there are too many parameters to learn which makes the learning problem computationally intractable. Hence, IBM Model 2 makes the following independence assumptions, :

- The length of $\mathbf{f}$ follows a uniform distribution.

- The translation probability (t) of a word at position j in $\mathbf{f}$ depends only on the words f_j and e_i.
- The alignment probability (a) of a word pair depends only the positions i, j and sentence lengths l, m in sentence $\mathbf{e}$ and $\mathbf{f}$ respectively.

The translation model for Model 2 is defined by Equation 3.2

$$Pr(\mathbf{f}, \mathbf{a}|\mathbf{e}) = \varepsilon \prod_{j=1}^{m} t(f_i|e_{a_j}) a(a_j|j, m, l) \tag{3.2}$$

where,
t: word translation probabilities
a: alignment probabilities

Reordering

The alignment probability a represents the reordering parameters in this generative process. The following observations can be made about Model 2's reordering model:

- The alignment probabilities are parameterized by the absolute positions of the words and the sentence lengths, which leads to a rather sparse model. In practice, this is alleviated to some extent by making f_j depend only e_i and l. Sparseness is further reduced by defining a in terms of relative alignment positions as opposed to absolute positions [ON03], as shown below. r is any suitably chosen function.

$$a(i|j, l) = \frac{r(i - j\frac{l}{m})}{\sum_{i'=1}^{l} r(i' - j\frac{l}{m})}$$

- The alignment probabilities, along with word transition probabilities, should be able to capture reorderings that are very frequent in corpora and have a fixed pattern in nature. Local head-modifier relations would typically belong to this category.
- The alignment of every word is independent of other words in the sentence, hence Model 2 does not capture the tendency of words in a single phrase to move together. Movement of entire phrases cannot be modeled.

3.1.2 HMM based model

The reordering of a word is generally not independent of other words in the sentence, especially it depends on words in the same phrase. This intuition is captured in the Hidden Markov Model (HMM) [VNT96], which conditions the alignment of position j in $\mathbf{f}$ on the alignment of the previous position,

$(j-1)$. Because of this first order dependence, it is referred to as an HMM model as compared to IBM Model 2, which can be said to be zeroth order HMM. The HMM model is defined as follows:

$$Pr(\mathbf{f}, \mathbf{a}|\mathbf{e}) = \prod_{j=1}^{m} t(f_i|e_{a_j})a(a_j|a_{j-1}, m, l) \tag{3.3}$$

where,
t: word translation probabilities
a: alignment probabilities

Reordering

The HMM model too suffers from sparsity problems. A solution similar to the one in Model 2 is suggested. The other deficiencies from alignment based models also carry over. The advantage of the HMM model over Model 2 is that it can potentially capture reordering of a sequence of words.

3.1.3 Fertility based models

Alignment based models do not address a very important issue *viz.* the word in the language **E** determines how many words in language **F** are generated. The number of words $\phi(e)$ generated by the word e is known as the *fertility* of e. A different generative process is used to incorporate fertility in the translation model, resulting in a different way of reordering. In the fertility based model, the translation occurs through the following process:

- Every word e_i corresponds to a *cept*. For every cept, decide the number of words ($\phi(e_i)$) to be generated.
- Generate the words in language F ($\tau_1^{\phi_i}$) for each cept in the sentence **e**.
- Generate ϕ_0 words in sentence **f** which do not have any correspondence to words in **e**.
- Permute the words generated in **f** to get the final translated sentence.

IBM Models 3 to 5 are specific instantiations of this generative process obtained by making specific independence assumptions.

IBM Model 3

IBM Model 3 can be represented as:

$$Pr(\mathbf{f}, \mathbf{a}|\mathbf{e}) \quad = \quad \binom{m-\phi_0}{\phi_0} p_0^{m-2\phi_0} p_1^{\phi_0} \times$$

$$\prod_{i=1}^{l} \phi_i!\; n(\phi_i|e_i) \times$$
$$\prod_{j=1}^{m} t(f_i|e_{a_j}) d(j|a_j, m, l) \tag{3.4}$$

where,
n: fertility probabilities, depending only word e
d: distortion probabilities, depending only on $a_j, m \text{and} l$
t: word translation probabilities
p_1: probability of generating an unaligned word

Reordering

The reordering is captured by the distortion probabilities, $d(j|a_j)$. This parameterization is again sparse, and does not take into account the tendency of phrases to move together in groups. With respect to reordering expressiveness, Model 3 is on par with Model 2, the only difference being the generative process. There are a few differences though which have implications for the reordering model:

- Model 3 allows one word from **e** to be aligned to multiple words in **f**.
- It does not allow one word from **f** to be aligned to multiple words in **e**.
- Moreover, the model allows more than one word to occupy the same position in sentence **f**. Allowing such invalid sentences wastes probability mass, hence Model 3 is referred to as a *deficient model*.

Thus, most of the limitations of alignment based models in modeling reordering apply. However, the generative process represents a significant step ahead in being able to clearly identify words generated from a single word on the source side. Model 4 makes use of this property.

IBM Model 4

Model 4 addresses the limitations of Model 3 by replacing absolute distortion with *relative distortion*. It makes the reasonable assumption that:

- Words generated from the same source word move together.
- Where a word aligns to depends on where the current word aligns to.

These assumptions are captured by the following generative process to explain the distortion:

- Every cept that has a fertility of at least 1, first generates the head of the cept. The head of a cept is defined to the word in the cept for

which the position is the smallest in **f**. The position of the head in **f** is chosen dependent on the position of the center of the previous cept in **f**.

- For all other words in the cept, the position is chosen relative to the head the cept.

Note that the notion of **head** is not the same as the linguistic notion of head, but is intended to capture the intuition that groups of words move together.

The distortion probabilities in Model 4 can thus be captured by one of the following parameters:
For the head of the cept, the distortion function would be:

$$d_1(j - \odot_{i-1} | A(e_{[i-1]}), B(f_j)))$$

For the other words of the cept, the distortion function would be:

$$d_{>1}(j - \pi_{[i]k-1} | B(f_j)))$$

where, A and B are word classes.

Complete lexicalization of the distortion parameters based on words would make the model very sparse, whereas parameterizing it based only on relative position would make it independent of the actual content of the sentence (as observed in the refinements to Model 2 and the HMM model). Model 4 therefore introduces the notion of *word classes*. Word classes would be POS tags, chunk tags, word clusters, etc. - any generalization that can group words into larger classes meaningful for alignment. The distortion parameters are conditioned on relative distance and the word classes, thus achieving a balance between lexicalization and non-sparsity of parameter space.

Thus, Model 4 helps to capture movements of groups of words. While Model 4 predicts the distance between words in language F, the HMM model predicts the distance between words in language E. The two models together thus capture localities in both languages. Hence the so-called *Model 6* [ON03] combines these two models using a log-linear framework.

Both Model 3 and Model 4 are deficient, thus wasting distortion probability mass. Model 5 fixes this issue by keeping track of the vacant position. Hence, Model 5 is the most sophisticated word based model.

3.2 Phrase based models

Words may not be the best units for translation. Consider the following:

- A word may have multiple translations which may be disambiguated by neighbouring words.

- A word may translate into more than one word.
- A word and its neighbours may be reordered together

It may make more sense to translate groups of words (phrases) as a single unit. This is the motivation behind the phrase based SMT model (PSMT). In PSMT, *phrase* refers to a sequence of contiguous words and should not be confused with the linguistic definition of *phrase*. The probability distribution for a sentence translation can be written as:

$$P(\mathbf{f}|\mathbf{e}) = \prod_{i=1}^{l} \phi(\overline{f}_i|\overline{e}_i) d(start_i - end_{i-1} - 1) \tag{3.5}$$

where,
$\overline{f}_i$ and $\overline{e}_i$ are phrases of the sentences $\mathbf{e}$ and $\mathbf{f}$ respectively.
ϕ is phrase translation probaility
d is the distortion probability, which given the probability of displacement of the current phrase relative to the previous phrase in $\mathbf{e}$.

3.2.1 Learning Phrase based models

The most popular approach to learning phrase based models in the alignment template approach [ON04]. This method first learns the word alignment model. The training data is aligned in both directions using the word based models. Phrases are then extracted from the alignment matrix based on consistency of alignment and expansion to unaligned words. Figure 3.2 shows an alignment matrix generated from word based models. Some of the phrase translations extracted from the matrix are:

- Chief Minister : मुख्य मंत्री
- of Maharashtra : महाराष्ट्र के
- Chief Minister of Maharashtra : महाराष्ट्र के मुख्य मंत्री

3.2.2 Reordering Phrase based models

In the alignment template approach, phrase tables are learnt from the alignment matrix. However, no reordering model is learnt over phrases. Hence, reordering in phrase based SMT is a problem. The default solution is to use a distance based distortion penalty, that penalizes non-monotone orderings. However, this is good only for languages pair which follow the same word order.

[Til04] has proposed a method to learn orientation of a phrase relative to the previous phrase. Here, they assume three orientation operations that can be applied on a phrase:

- Monotone: The phrase follows the previous phrase in monotone ordering i.e. no reordering occurs.

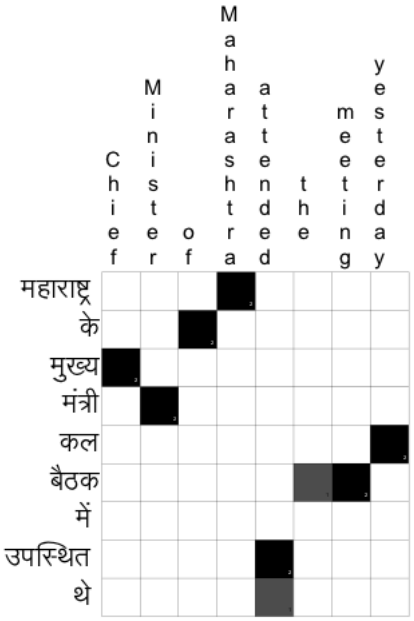

Figure 3.2: Alignment matrix to learn phrase table

- Swap: The phrase is swapped with the previous phrase in the translation.
- Discontinuous: The phrase could get reordered to any position in the target language sentence.

From the alignment matrix, it is easy to count the occurences of each orientation type for every phrase pair in the corpus:

- An alignment point to top-left indicates a monotone order.
- An alignment point to the bottom-right indicates a swap.
- No alignment to the top-left or bottom right indicates a discontinuous orientation.

The maximal likelihood estimate for the lexicalized orientation probability of a phrase pair, with smoothing, is given by:

$$Pr(\text{orientation} = o|\overline{f}, \overline{e}) = \frac{\sigma\ Pr(orientation) + \text{count}(\text{orientation}, \overline{e}, \overline{f})}{\sigma + \sum_o \text{count}(o, \overline{e}, \overline{f})} \tag{3.6}$$

where,
o is the orientation which can take one of three values monotone, swap, discontinuous.

$Pr(orientation)$ is the prior probability of the orientation, which can be easily computed from the orientation statistics.

Lexicalized reordering has been shown to be better than (i) monotone reordering and (ii) allowing the decoder to swap adjacent phrases. The model is heavily lexicalized, and faces sparsity problems since the orientation is learnt for every phrase pair. The smoothing with prior orientations alleviates the problem to some extent.

3.3 Syntax based Models

Phrase based systems represent a good advance over word based models in terms of learning translations, but they haven't advanced technology in reordering models. PSMT exhibits the following limitations:

- It can learn local reorderings only on account of memorizing phrases
- The alignment template based training procedure makes it difficult to learn a reordering model in a systematic manner
- The reorderings that are learnt are highly lexicalized - in case of both the phrase table and the lexicalized reordering method. It is not possible to learn general rules of reordering in the PSMT framework.

The key point is that phrase based and word based SMT models are not able to learn general reordering rules because they make no use of syntactic knowledge. Hence the expressive power and generalization capability required to capture reorderings is very limited. The central idea in Syntax based SMT models is that syntactic representation provides a powerful representation language to learn reordering rules.

[YK01] proposed a generative model for tree-to-string translation. In this case, a source parse tree is tranformed into a target string in the noisy channel through the following operations:

- Reorder children of a node, to explain reordering
- Insert nodes to the right or left of a node. This explains words in the target sentence, which have no equivalent in the source string.
- Translate the leaf node words to target language

Reading off the leaf nodes gives the target language sentence. Figure 3.3 illustrates the process with a English-Japanese translation.

The authors make a couple of independence assumptions in order to simplify the model:

- The tranformations at each node are independent of other nodes.

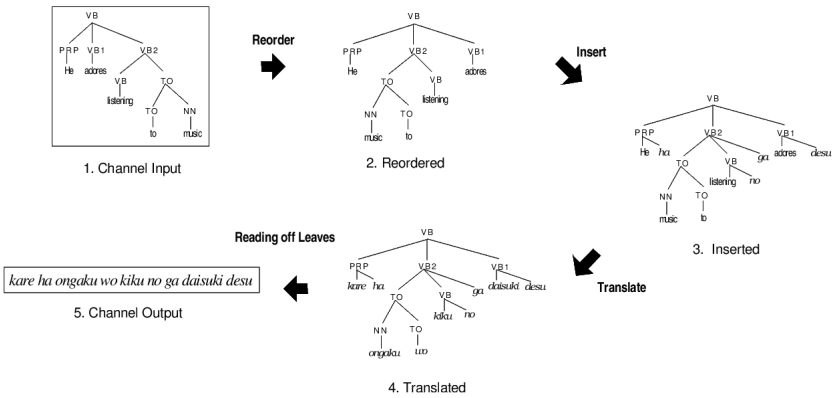

Figure 3.3: Syntax based SMT system [Src: [YK01]]

- Each tranformation is independent of the other transformations in that node.

With these assumptions the translation model can be represented as:

$$Pr(\mathrm{f}|\epsilon) = \sum_{\theta:Str(\theta(\epsilon))=\mathrm{f}} Pr(\theta|\epsilon) \tag{3.7}$$

where,
ϵ = input parse tree
θ: sequence of parameters $\theta_1, \theta_2, \ldots \theta_n$ for each node of the parse tree.
Each θ_i is a triple of parameters μ_i, ρ_i, τ_i for the word insertion, reordering and translation probabilites at node i
The marginalization is over all parse tree transformations that can generate the same target string.

The reordering parameter in this model is of the form

original sequence of non-terminals $\rightarrow$ reordered sequence <probability>

Figure 3.4 illustrates the reordering parameters.

So, syntax based methods learn linguistic rules for reordering and thus provide a higher level of generalization than word and phrase based models. [YK01] show that the syntax based approach produces better alignments than IBM Model 5. Rules learnt by syntax based models can learn constituent order divergences.

However, this model is limited to learn divergences that are found under the same node of the tree. If the divergences involve long distance dependencies, this model will not be able to find it. [QMC05] propose a model based on dependency tree to string translation that can use a dependency parse to handle reordering involving long distance dependencies.

original order	reordering	P(reorder)
PRP VB1 VB2	PRP VB1 VB2	0.074
	PRP VB2 VB1	0.723
	VB1 PRP VB2	0.061
	VB1 VB2 PRP	0.037
	VB2 PRP VB1	0.083
	VB2 VB1 PRP	0.021
VB TO	VB TO	0.251
	TO VB	0.749
TO NN	TO NN	0.107
	NN TO	0.893
⋮	⋮	⋮

Figure 3.4: Reordering parameters in syntax-based SMT [Src: [YK01]]

Chapter 4

Handling reordering during decoding

The task of decoding in machine translation can be explained thus: Given a source language sentence, use the available information (language model, translation model, other features) to construct a target language sentence. The decoder is responsible for generating the target language words in the correct order. The idea is to select the translation which maximizes the score/minimizes the penalty as defined by the translation models. Decoding is typically cast as a problem of *searching* through multiple candidate translations by progressively building partial translations.

As compared to inference in scalar classification, decoding in machine translation is a very challenging inference task. This complexity results from the fact that the space of possible translations for each input sentence is exponential in the sentence length. Consider a sentence of length n, with each word having m possible translations. The number of bag-of-words is m^n. Reordering causes further explosion in the search space since $n!$ permutations are possible for each possible bag-of-words, giving a total of $m^n\ n!$ translation candidates. Indeed, [Kni99] has shown that the decoding problem is NP-complete even for simpler settings.

The reorderings contribute in a great way to the intractability of decoding. In this chapter, we discuss the methods to constrain possible reorderings in order to make decoding efficient. At the same time, it is important to ensure that good translation candidates are evaluated during decoding. We describe how prior information about reordering can be provided to the decoder.

4.1 Search space for a decoder

The decoding problem is cast as a search problem. The decoder builds the sentence translation progressively, one word or phrase at a time. Thus the

search space consists of *partial translations (hypotheses)*, each of which is a node in the search graph.

Since the target sentence is reordered with respect to the source sentence, the decoder does not pick source words in source order for translation. Instead, words in the source sentence may be picked up for translation in any order, while the target sentence is constructed sequentially. *A hypothesis consists of information about which source sentence words have been translated, the current position in the source sentence and the partially translated target sentence.*

Given a hypothesis, the next hypothesis can be obtained by expanding the partial translation. This involves choosing a source language position to translate and choosing the target language word. A hypothesis and its successor are connected by an edge in the search graph.

It is possible to combine multiple hypotheses into a single hypothesis [Koe08]. Two hypotheses can be combined if the same input source positions have been covered and the last (n-1) words of the partial translations using a n-gram translation model are identical. In this case, only the hypothesis with the higher score needs to be retained, since any complete translation containing a hypothesis with a lower score can be replaced by higher scoring equivalent hypothesis.

This is illustrated in the following example of Hindi to English translation. Hypotheses H1 and H2 are partial translation of the sentence S, with words 1, 3,4 and 5 translated. The last word in both the sentences is *going*. If a bigram language model is used, then expanding only the better of the two hypotheses is sufficient.

S: मै बाज़ार जा रहा था
H1: I was going
H2: We were going

Hypothesis recombination results in a search graph which is a lattice.

4.2 Stack based decoding

Stack based decoding using beam search is a very common decoding technique for word and phrase based SMT models. The algorithm basically builds a translation progressively by searching the translation lattice described in the previous section. Stack decoding proceeds as follows:

1. For maintaining hypotheses (partial translations), priority queues (referred to as stack in literature, hence we will use that term henceforth) are used. There is one priority queue for every possible length of the partial translation.

2. Decoding starts with an empty hypothesis i.e. no words translated. The hypothesis is expanded by choosing an source sentence position and possible translations for the source word at that position. These hypotheses are pushed on to the appropriate stack (depending on length of the hypothesis).

3. This kick starts the iterative operation of the decoder.

4. In every iteration, it selects the best hypothesis from one of the stacks. The hypothesis score is computed using the translation and language models (or log-linear model).

5. This hypothesis is expanded by choosing the next source word position to translate. All resulting hypotheses are again added to the appropriate stack. This iterative process continues, as long as all hypotheses are not evaluated.

6. In order to avoid evaluating too many hypotheses, the stacks are pruned periodically. Pruning could be based on a limit on the number of hypotheses per stack (histogram pruning) or a threshold on the hypothesis score (threshold pruning).

7. Since the search space is a lattice, every new hypothesis should be checked for recombination with an existing hypothesis.

4.3 Reordering Constraints

As seen in the previous section, the decoder expands hypotheses by consuming the source language word in different permutations. However, this will result in an exponential number of hypotheses. Therefore, it is necessary to restrict the permissible reorderings. Two reordering constraints have been widely used - the IBM constraint and the ITG constraint [ZN03].

4.3.1 IBM constraint

The IBM constraint is a distance based constraint. It places a limit on the number of uncovered words on the source side that can be aligned to the target word position being considered. Generally, the limit is set to $k = 4$, yielding the number of possible reorderings explored would be $O(4^n)$. Therefore, decoding with IBM constraint will be able to perform short range reorderings. Larger distance dependencies are not uncommon in language, hence the IBM constraint severely constrain the search space. In Figure 4.1, the square boxes represent the uncovered source positions that can be expanded.

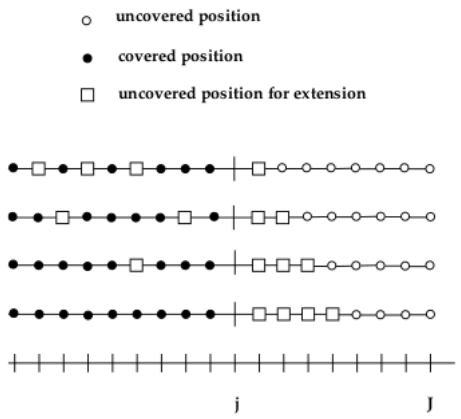

Figure 4.1: IBM constraints [Src: [ZN03]]

4.3.2 ITG constraint

The ITG constraint takes a different approach to reordering. Instead of reordering at word level, it allows reorderings at block level. A block is a sequence of words on both the source and target sides. In the ITG constraints, words in a block are translated in a monotone order, but adjacent blocks may be swapped with each other. Search using the ITG constraints can be done in polynomial time to find the best possible reordering using a dynamic programming algorithm. The limit on the number of reorderings can be shown to be around $O(6^n)$.

So ITG constraints cover larger number of reorderings. ITG constraints thus allow reordering over a larger distance, as long as adjacent blocks are being swapped. The constraints capture the general observation that phrases move together as a whole and entire phrases could be swapped during translation. However, the ITG constraints cannot capture reordering involving noncontiguous phrases. Figure 4.2 shows the block reorderings possible with ITG constraints.

4.3.3 Clause boundary constraint

As mentioned in Chapter 2, reordering does not generally cross clause boundaries. This provides a natural constraint for decoding. [RBV+11] experimented with clause boundariers as barriers across which reorderings could not take place. They found improvements in BLEU score as well as in human judgment for sentences involving only finite clauses. For non-finite clauses, clause boundaries did not provide any improvement. This is evident from the examples in Chapter 2, where translations of non-finite clauses did not respect clause boundaries.

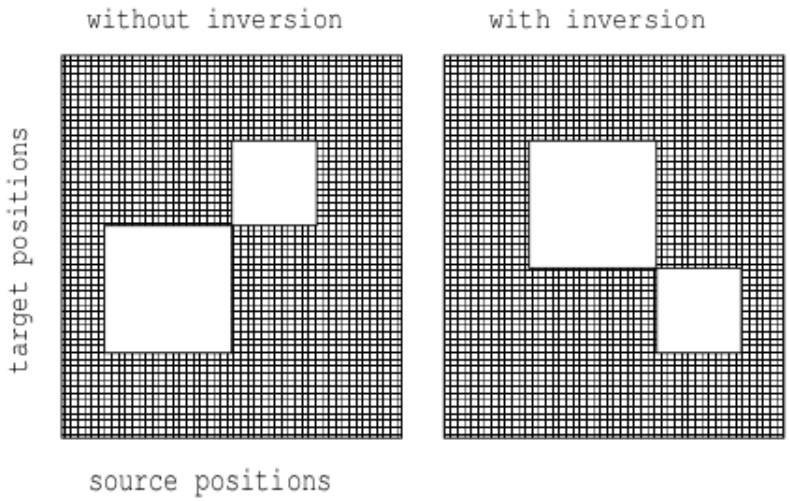

Figure 4.2: ITG constraints [Src: [ZN03]]

4.4 Enumerating good hypotheses

Applying reordering constraints is a way of reducing the search space. However, we need to ensure that good hypotheses are not lost due to these restrictions. Sometimes, we may have an idea of potentially good reorderings. This is possible when source side reordering is done (discussed in Chapter 5). In such cases, these hypotheses can be enumerated and the decoder can be explicitly asked to consider these hypotheses during search. This will ensure that good hypotheses are expanded during decoding. One way of detecting potential good hypotheses is to use the source reordering techniques discussed in Chapter 5.

Chapter 5

Pre-processing and post-processing to improve reordering

The reordering techniques discussed in the previous chapters make use of generative probabilistic models. However, these models make many independence assumptions and approximations for computational reasons and for limiting the use of linguistic knowledge. Therefore, they are far from being able to exploit all available knowledge.

Linguistic and other sources of knowledge can be incorporated in a preprocessing step to generate input for the SMT system which conforms to the assumptions made by the system. This will make it easier to model the data with current SMT models. In this chapter, we discuss one such preprocessing technique which involves *reordering* the source language sentence to conform to the target language word order. This approach has been widely used in phrase-based SMT (PSMT).

On the other hand, the computational complexity involved in searching for a translation during decoding prohibits expensive and deep analysis. Therefore, there is a case for *re-ranking* the top-k candidate translations obtained from decoding by doing a more deeper analysis of these candidates with a richer set of features. We describe this method too in this chapter.

5.1 Source side reordering

5.1.1 Need for source side reordering

SMT systems exhibit the following deficiencies with respect to handling reordering:

- Reordering constraints are applied to restrict the decoder's search space, which is exponential in sentence length. These constraints are

generally distance based, and typically result in translation candidates involving long distance reorderings being discarded.

- It is difficult to learn movements of large phrasal units, since the reordering models are very lexicalized. There wouldn't be enough support in the corpus for entire phrases in the absence of any linguistic generalization.

- In the case of alignment template based PSMT model, no phrase level alignment model is directly learnt from the corpus. The phrase table is generated from word alignments, while no phrase level alignment is learnt from the word alignment matrix. So the reordering model is PSMT is weak and the methods suggested in Chapter 3 are only tweaks to make up for fundamental weaknesses in PSMT with respect to modeling reordering.

- PSMT cannot learnt noncontiguous phrases.

5.1.2 How does source side reordering help?

Many of the problems mentioned in the previous section can be overcome using source side reordering of sentences to conform to the word order of the target language. Phrase based SMT systems benefit the most from this pre-processing. Such a reordering helps for the following reasons:

- Pre-processing generates a source sentence whose word order matches the target language word order before the decoder starts its search. The decoder can now do a better translation by looking at reorderings in a small window since the long distance dependencies can be handled by the pre-processing step. So, it can get around restrictions imposed by reordering constraints. Example 5.1.2 shows how monotone correspondences are established between source and target language phrases, simplifying the decoder's search. Generally, long distance constituent-order divergences will be benefitted through pre-preprocessing.

- In PSMT, local reorderings can be lexicalized by learning phrase translations from word level alignments. Pre-processing aids this process by making previously noncontiguous phrases contiguous, enabling learning of better phrase table entries. Source side reordering thus enhances the strength of phrase based systems, *viz.* being able to learn longer phrases - doing some local reordering, context disambiguation, and agreement in the process. In Example 5.1.2, because of pre-processing, the phrase table can contain सुबह एक बैठक थी, which naturally captures the verb-adverb reordering for this particular verb-adverb pair.

Pre-processing thus helped learn word reordering in a noncontiguous phrase.

Thus, if the two languages have the same word order, even a distance penalty based decoder can give good translations. [DZ07] also confirms through experiments that the above mentioned factors contribute to the improvement in the performance of a PSMT system.

	S	V	O	PP_1	PP_2
S	I	had	a meeting	with the manager	in the morning
S_r	I	with the manager	in the morning	a meeting	had
	S	PP_1	PP_2	O	V
T	मेरी	मैनेजर के साथ	सुबह	एक बैठक	थी

where,
S: source sentence
S_r: reordering source sentence after pre-processing
T: target language sentence

5.1.3 Reordering with hand coded rules

The most straightforward way to incorporate source side reordering is to use handcoded rules on source sentences before training or decoding. [CKK05] and [AHBS08] have taken this approach to handle the most important word order divergences for the German-English and Hindi-English language pairs respectively. [CKK05]'s rules handle movement of finite, infinite verbs, subject, particles and negation modifiers. [AHBS08]'s rules are intended to capture the following fundamental reordering divergence pattern between English and Hindi,

$$SS_mVV_mOO_mC_m \leftrightarrow C'_mS'_mS'V'_mV'O'_mO'$$

where,
S: subject
V: verb
O: object
C_m: clause modifier
X': corresponding constituent in Hindi where X is S, O or V
X_m: modifier of X

Use of source reordering has been shown to improve performance of phrase based SMT systems. Language analysis resources and tools are required only on the source language side.

On the downside, language pair specific rules will have to be written. Handcoding patterns is difficult for many ambiguous reorderings like movement of infinitive clauses or long distance dependencies. Source reordering

represents a hard constraint for the decoder. The decoder would have evidence from multiple sources to disambiguate reorderings, but it will find it difficult to ignore the hard constraint. This would be concern if the reordering is wrong due to parsing errors, etc. Hence, an early commitment to reordering through source ordering should be made only in the case of very definite and unambiguous reordering patterns.

5.1.4 Reordering with automatically extracted rules

Instead of handcoding the reordering rules, there have been attempts to learn reordering patterns using various linguistic resources. These attempts have tried to learn both lexicalized and unlexicalized rules.

Learn rules from parse trees

[XM04] uses parse trees on both source and target language sides to learn context free reordering patterns of the form:

$$(SrcRule, TgtRule, SrcHeadPos, TgtHeadPos, ChildAlign)$$

where,
$SrcRule$ is a context free rule on the source side.
$TgtRule$ is a corresponding context free rule on the target side.
$SrcHeadPos$ is an integer giving the position of the head of $SrcRule$ in the source sequence.
$TgtHeadPos$ is an integer giving the position of the head of $TgtRule$ in the target sequence.
$ChildAlign$ gives a correspondence between the source side and target side nodes in the rules.
A example of such a rule would be:

$$(NP \rightarrow NP_1 NP_2, NP \rightarrow NP_2 NP_1, 0, 1, 0:1, 1:0)$$

- To learn these rules, sentence pairs in the training corpus are aligned using a word aligner.
- Nodes in the parse tree which cover the same subset of aligned words are linked to each other, and rewrite patterns are read off the parse tree from linked nodes which have the same structure (same head and siblings).
- To keep a limit on the number of rules, certain support criteria are used to prune away unwanted rules.
- The rules are then clustered into groups by the source node.

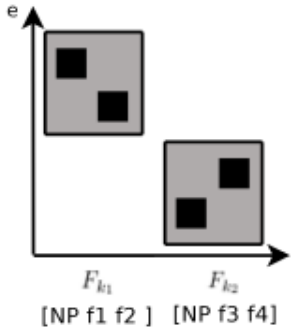

Figure 5.1: Learning Reordering patterns using chunks and word alignments

- These groups are then organized into a hierarchy of specific to general rules. The more lexicalized a rule is, the more specific it is.
- Before applying phrase based SMT training or decoding, the input sentences are reordered by looking up the rules in this hierarchy and applying the most specific rules.

Most of the rules learnt are lexicalized, demonstrating the importance of lexicalization of rules. The authors have a reported 10% improvement in BLEU score, but the downside is that this approach requires parsers for both the languages involved.

Learn rules from chunk/POS tags

[ZZN07] learn rules involving chunk and POS tags of the form:

> *Sequence of chunk tags → sequence of reordered position of source chunks*

For example,

$$NP_1\ NP_2 \rightarrow 1\ \ 0$$

The rules indicate the reordering of words that takes place for the chunk/POS tag pattern on the LHS.

To learn these rules, a standard template based aligner is first used to extract phrase translation pairs. From the word-to-word alignment, a chunk-to-word alignment rule is obtained by mapping the chunk labels spanned by the phrase to the aligned word positions. Figure 5.1 shows how the pattern shown above could be learnt from word alignments. [CM06] also suggest a similar approach to extract rules over POS tags only. These methods give a 0.5 to 2 point improvement in the BLEU score. The chunk based reordering gives better improvements compared to POS tag based reordering.

5.2 Re-ranking of translation hypothesis

5.2.1 A discriminative approach to translation

In a generative framework, we are not able to integrate arbitrary features to score translation candidates. This limits the use of linguistic cues for scoring translations. Hence, [ON01] proposed a discriminative model for scoring translations, using a log-linear combination of arbitrary features. Thus, the score for a sentence pair (s, t) would be

$$f(s,t) = \exp \sum_i \lambda_i \psi_i(s,t)$$

The best translation for a sentence s is given by

$$t^* = \arg\max_t f(s,t)$$

However, given the large search space of possible translations, most of the state-of-the-art SMT systems do not perform complete discriminative training. Rather, individual features are learnt using the generative techniques discussed in the previous chapters and then the features are combined using the log-linear model. Parameter tuning techniques like MERT [Och03] are used to learn the feature weights. Some of the common features incorporated are the phrase translation model scores, lexical model scores, distortion model scores, language model scores, word length penalty, etc.

5.2.2 Why re-rank candidate translations?

Though it is possible to integrate a large number of features to get a richer translation model, this places a lot of computational demand on the decoding stage, which has to search through a large hypothesis space. The compromise is to first extract the top-k translation candidates during the decoding stage using a small number of features like the ones mentioned above. These top-k translation candidates are then re-ranked using a larger set of features to select the best translation.

This means that the decoder should be able to extract the k-best translations for the source sentence. This can be done by maintaining the hypotheses in a structure called the 'word graph' during decoding. The word graph is basically a translation lattice with scores of partial translations attached. Once the word graph is constructed after decoding, the future cost estimates at each node can be set to be equal to the translation model score starting at that node. In this case, it is optimal to run A^* search on this graph, which can be used to retrieve the top-k candidate translations [UON02].

5.2.3 Features for re-ranking

The Syntax for Statistical Machine Translation Workshop, 2003 [OGK+03] experimented with a lot of features for discriminative training of SMT models. We list down a few of the features that we believe could be useful for identifying candidate translations that are well ordered.

- POS tag language model for the target language. The POS tag sequence will capture local syntactic sequence, and was observed to perform very well.
- Parser scores: Target side parser scores could indicate how grammatical the candidate translation is. This could be seen as a comment on the correctness of the reordering.
- A lot of boolean feature functions based on testing syntax could be developed:
  - Are the number of arguments to the verb correct?
  - Are different constituents in correct relative positions? e.g. in an SOV language, does the verb follow the object?
  - Position test features like: does the sentence start with a noun?
  - Are the occurrences of conjunctions balanced on both sides? Balance is determined by whether within the boundary of a constituent, the sections before and after a conjunction word end with the same or similar POS tags.

Chapter 6

Conclusion and Future work

6.1 Summary and Observations

What divergences patterns can be captured by SMT models?

- Local reorderings like head-modifiers can be done by current SMT models. IBM Model 4 and the HMM model learn sophisticated distortion models which are able to learn movement of sequence of words.
- Though the reordering model in phrase based models is weak, they are still able to do lexicalized reordering as they simple memorize phrase translation pairs which incorporate reorderings.
- Constituent parse based systems are able to learn general linguistic reordering rules, and are better positioned to handle more high level reordering phenomena like constituent order. Similarly, source reordering systems can also handle these phenomena. In fact, there are similarities in the methods used by syntax based SMT systems and syntax based source reordering systems to learn reordering rules. The difference is in the relation of the reordering component to the rest of the SMT system. Syntax based systems integrate the reordering component into a unified stochastic system. This is a more elegant design and is at the cutting edge of research. Reordering system try to retrofit existing and proven SMT systems like PSMT with syntactic knowledge by doing preprocessing.
- Most SMT systems cannot handle reorderings involving noncontiguous phrases or long distance dependencies. Dependency based SMT systems or reranking systems using intelligent dependency based features hold a promise for translating such sentences.

Role of linguistic knowledge for reordering

It is observed that with increasing linguistic knowledge, the reordering performance of SMT systems improve. The linguistic knowledge gives a couple of benefits. First, the simpler models just do not have the expressive power to learn and represent reordering patterns. Linguistic representation in the form of CFG, POS, chunk knowledge, etc. allows SMT models to generalize from the data and go beyond learning lexicalized patterns. Secondly, linguistic knowledge plays an important role in providing efficient solutions to inherently intractable problems. For instance, knowledge of reordering patterns can help design better search heuristics for reordering.

Then, why use statistical methods?

- Uncertainty and ambiguity are a fact of life in NLP problems, including reordering. Rule based system would become too unwieldy trying to solve these issues. A statistical framework provides a principled way to looking at data and knowledge sources to make a informed decision.
- The traditional method of using linguistic knowledge was to having experts develop reordering rules. Statistical methods offer ways to learn reordering rules from data, thus reducing expert involvement.
- Looking for the best reordering is a search problem. In a symbolic processing system, it would be cast as a combinatorial search problem. In a statistical system, the problem can be cast as a numerical optimization problem, which is easier to solve than combinatorial optimization problems.

Key principles in design of reordering solutions

- The translation model should be expressive enough to score reorderings properly. The use of linguistic knowledge here is very useful.
- Decoding is an NP-complete problem, and the key is to finding a good solution is to design good search heuristics.

6.2 Future Directions

- For a good translation, it is necessary that the decoder is able to evaluate good hypotheses. This depends on the having a good set of heuristics to navigate the translation search space. We would like to explore heuristics motivated by linguistic knowledge for navigating the translation search space.

- Generative modeling of translation is not able to exploit linguistic information. Hence, discriminative models have become popular lately. It would be interesting to see how discriminative models can be learnt for reranking top-k candidate translations to get the best reordering.
- Many of the best performing reordering methods rely on the presence of linguistic resources like constituency and dependency parsers. These tools are not available for most Indian languages. We would be interested in exploring how reordering can be done in such resource scarce circumstances.

Bibliography

[AHBS08] R. Ananthakrishnan, Jayprasad Hegde, Pushpak Bhattacharyya, and M. Sasikumar. Simple Syntactic and Morphological Processing Can Help English-Hindi Statistical Machine Translation. In *IJCNLP*, 2008.

[BPPM93] Peter F. Brown, Vincent J. Della Pietra, Stephen A. Della Pietra, and Robert L. Mercer. The mathematics of statistical machine translation: parameter estimation. *Computational Linguistics*, 1993.

[CKK05] Michael Collins, Philipp Koehn, and Ivona Kučerová. Clause restructuring for statistical machine translation. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics - ACL '05*, 2005.

[CM06] Josep M Crego and Jose Marino. Integration of POStag-based source reordering into SMT decoding by an extended search graph. In *In Proceedings of AMTA*, 2006.

[Dor94] Bonnie J. Dorr. Machine Translation Divergences: A Formal Description and Proposed Solution. *Computational Linguistics*, 1994.

[DPB03] Shachi Dave, Jignashu Parikh, and Pushpak Bhattacharyya. Interlingua-based English Hindi Machine Translation and Language Divergence. *Machine Translation*, 2003.

[DZ07] Mark Dras and Simon Zwarts. Syntax-Based Word Reordering in Phrase-Based Statistical Machine Translation: Why Does it Work? In *MT Summit 2007*, 2007.

[Kni99] Kevin Knight. Decoding complexity in word-replacement translation models. *Computational Linguistics*, 1999.

[Koe08] Philipp Koehn. *Statistical Machine Translation*. Cambridge University Press, 2008.

[Och03] Franz Josef Och. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - ACL '03*, volume 1, pages 160–167, Morristown, NJ, USA, July 2003. Association for Computational Linguistics.

[OGK+03] Franz Josef Och, Daniel Gildea, Sanjeev Khudanpur, Anoop Sarkar, Kenji Yamada, Alex Fraser, Shankar Kumar, Libin Shen, David Smith, Katherine Eng, Viren Jain, Zhen Jin, and Dragomir Radev. Workshop on Syntax for Statistical Machine Translation. In *Workshop on Syntax for Statistical Machine Translation*, 2003.

[ON01] Franz Josef Och and Hermann Ney. Discriminative training and maximum entropy models for statistical machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics - ACL '02*, page 295, Morristown, NJ, USA, July 2001. Association for Computational Linguistics.

[ON03] Franz Josef Och and Hermann Ney. A Systematic Comparison of Various Statistical Alignment Models. *Computational Linguistics*, 2003.

[ON04] Franz Josef Och and Hermann Ney. The Alignment Template Approach to Statistical Machine Translation. *Computational Linguistics*, 2004.

[QMC05] Chris Quirk, Arul Menezes, and Colin Cherry. Dependency Treelet Translation: Syntactically Informed Phrasal SMT. In *Proceedings of ACL*, 2005.

[RBV+11] Ananthakrishnan Ramanathan, Pushpak Bhattacharyya, Karthik Visweswariah, Kushal Ladha, and Ankur Gandhe. Clause-Based Reordering Constraints to Improve Statistical Machine Translation. In *IJCNLP*, 2011.

[Sin03] Surajbhan Singh. *English-Hindi Translation Grammar*. Prabhat Prakashan, 2003.

[ST05] R Mahesh K Sinha and Anil Thakur. Translation Divergence in English-Hindi MT. 2005.

[Til04] Christoph Tillmann. A unigram orientation model for statistical machine translation. In *Proceedings of HLT-NAACL*, 2004.

[UON02] N. Ueffing, F. J. Och, and H. Ney. Generation of word graphs in statistical machine translation. In *In Proc. Conf. on Empirical Methods for Natural Language Processing.*, 2002.

[VNT96] Stephan Vogel, Hermann Ney, and Christoph Tillmann. HMM-based word alignment in statistical translation. 1996.

[XM04] Fei Xia and Michael McCord. Improving a statistical MT system with automatically learned rewrite patterns. In *Proceedings of the 20th International Conference on Computational Linguistics - COLING '04*, 2004.

[YK01] Kenji Yamada and Kevin Knight. A syntax-based statistical translation model. In *Proceedings of the 39th Annual Meeting on Association for Computational Linguistics - ACL '01*, 2001.

[ZN03] Richard Zens and Hermann Ney. A comparative study on reordering constraints in statistical machine translation. In *In Proc. Conf. on Association of Computational Linguistics.*, 2003.

[ZZN07] Yuqi Zhang, Richard Zens, and Hermann Ney. Chunk-Level Reordering of Source Language Sentences with Automatically Learned Rules for Statistical Machine Translation. In *Workshop on Syntax and Structure in Statistical Translation , NAACL-HLT*, 2007.