Reordering Models for Statistical Machine Translation: A Literature Survey

Piyush Dilip Dungarwal
123050083

June 19, 2014

In this survey, we briefly study various reordering models that are used with statistical translation models. Reordering model is one of the important component of any statistical machine translation system. Problem of reordering is NP-Hard itself. In this survey, we study various reordering approaches that can be used to solve this problem. We first study simple distortion-based reordering which is used with phrase-based and factor-based models. Next, we discuss limitations of this distance-based approach. Then we introduce a new source-reordering based approach to handle the reorderings based on structural information of the input text. We study how to use parse trees and shallow parsing for source-side reordering.

1 Distortion penalty-based reordering

We know noisy channel approach to translation:

$$e_{best} = argmax_e p(f|e) p_{LM}(e)$$

For phrase-based models, we decompose $p(f|e)$ into :

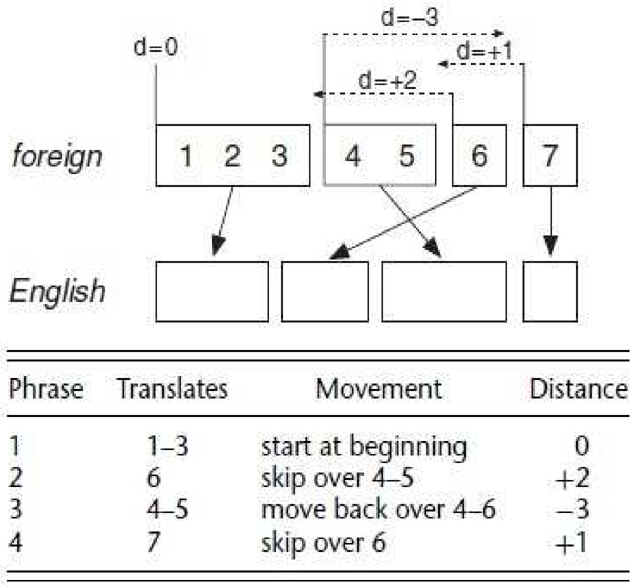
$$p(\bar{f}_1^I|\bar{e}_1^I) = \prod_{i=1}^{I} \phi(\bar{f}_i|\bar{e}_i) d(start_i - end_{i-1} - 1)$$

Foreign sentence f is broken into I phrases. Each foreign phrase $\bar{f}_i$ is translated into an English phrase $\bar{e}_i$.

Phrase-based model handles reordering by a distance-based reordering model. This distance is computed relative to the previous phrase. Consider this notation:

$start_i$ is the starting position of the foreign phrase that translates to ith English phrase. end_i is the ending position of the foreign phrase that translates to ith English phrase. Now, reordering distance is computes as: $start_i - end_{i-1} - 1$.

Reordering distance is nothin but the number of words skipped, while taking foreign words out of sequence. Figure 1 shows an example of distance-based reordering.



Phrase	Translates	Movement	Distance
1	1–3	start at beginning	0
2	6	skip over 4–5	+2
3	4–5	move back over 4–6	−3
4	7	skip over 6	+1

Figure 1: Example of distance-based reordering [Koehn, 2010]

Reordering probabilities d are estimated as: $d(x) = \alpha^{(|x|)}$. α is a parameter such that $\alpha \epsilon [0, 1]$, which helps reduce $d()$ to a probability distribution.

1.1 Lexicalized reordering

Instead of reordering based on the phrase positions, lexicalized reordering is conditioned on the actual phrases. There are three types of lexicalized reorderings: monotone (m), swap with previous phrase (s), and discontinuous (d).

Figure 2 shows three types of phrase orientations. We need to create a probability model which can predict the orientation type of the given phrase. We use word alignments for the same. Orientation type can be detected as follows:

- Monotone ordering (m): if a word alignment point to the top left exists

- Swap with previous phrase (s): if a word alignment point to the top right exists
- Discontinuous (d): if no word alignment point exists to the top left or to the top right

Now, reordering model p_o is learnt from counting number of phrases in training corpus having particular orientation. We use maximum likelihood principle:

$$p_o(orientation|\bar{f}, \bar{e}) = count(orientation, \bar{f}, \bar{e}) / \sum_o count(o, \bar{f}, \bar{e})$$

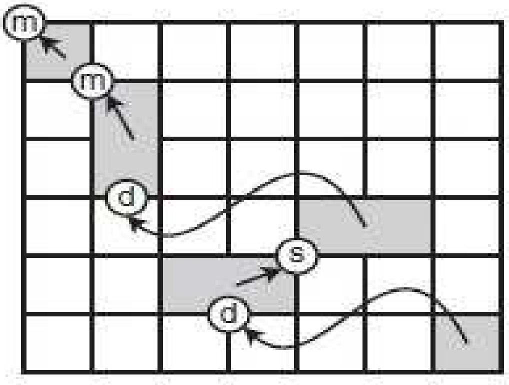


Figure 2: Example of lexicalized reordering [Koehn, 2010]

2 Source-side reordering

As we know, English follows SVO (Subject-Verb-Object) order while Hindi follows SOV (Subject-Object-Verb) order. Phrase-based model and Factored models try to handle this reordering phenomenon using a *distortion penalty.* A word is allowed to be placed anywhere inside a fixed-size window. But, if word goes outside this window, it leads to the inclusion of penalty while scoring the translation. This approach of handling reordering phenomenon performs well when ordering of words does not vary too much after translation. Hence, this approach is not suggested for translation between English and Indian languages in general.

Another useful appraoch for handling reordering is to convert source side sentence into a sentence in which words are ordered according to the position of the words in target side. This is called source-side reordering. This step needs to be done before actual training. We need to source-reorder training text as well as test sentences.

2.1 Parse tree-based reordering

Source-side reordering can be achieved by using syntactic parse tree of the source sentece. We either need to learn the reordering rules or need to find them out manually. A rich set of rules is created, which work on reordering children of nodes of a syntactic parse tree [Patel et al., 2013]. Source reordering helps to learn better word alignements and better phrase extraction.

Approach:

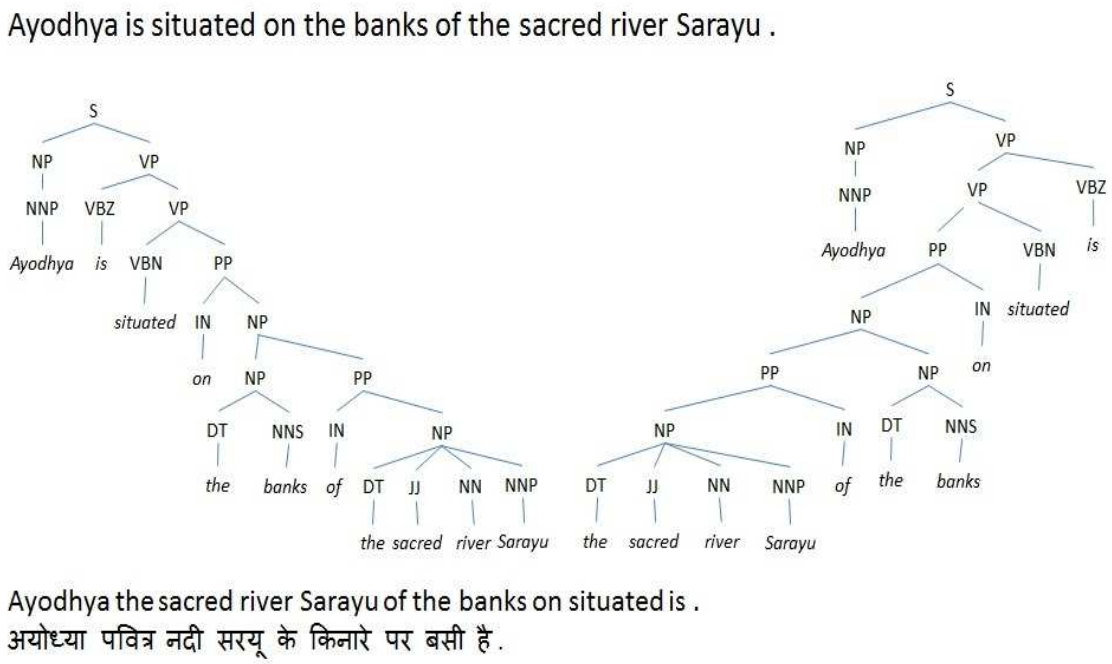


Figure 3: Examples of source-side reordering using parse tree

The source and the target sentences are manually analyzed to derive the tree transformation rules. From the generated set of rules we select rules which seem to be more generic. There are cases where more than one possible correct transformations for an English sentence can be found, as the target language (Hindi) is a free word order language within certain limits. In such cases word order close to English structure is preferred over possible word orders with respect to Hindi. There are 5 categories which are most prominent candidates for reordering. These include VPs (verb phrases), NPs (noun phrases), ADJPs (adjective phrase), PPs (preposition phrase) and ADVPs (adverb phrase).

2.2 Chunk-based reordering

While translating from source languages which dont have a constituency or dependency parser, it is very difficult to reorder the source sentence to match the word order of the target language sentence.

We can use shallow parsing techniques for source-side reordering. Chunk level tagging can be seen as intermediate annotations between POS tagging and parsing. The overall architecture of a translation system with chunk-based source reordering is shown in Figure 4.

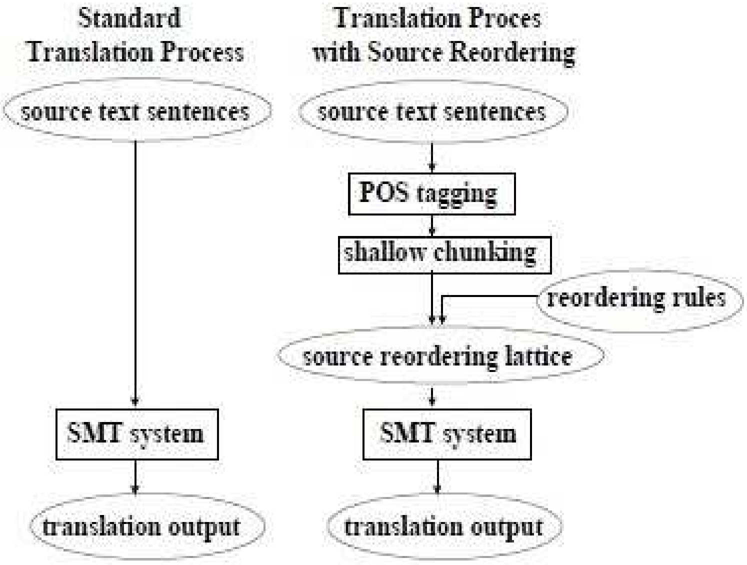
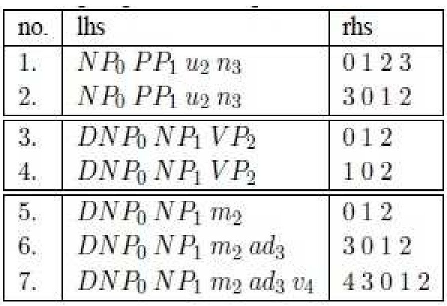


Figure 4: Architecture of a translation system with and without chunk-based source reordering [Zhang et al., 2007]

A reordering lattice is used for input to the translation system, instead of single sentence. Using lattice helps considering all possibilities of source-reordered input with their probabilistic scores. We first POS tag the input sentence and get chunk-level annotations. Then reordering rules are applied on these chunks to get a reordering lattice.

Learning reordering rules:

Reordering rules can be learnt from the training corpus or they can be formulated manually. Reordering rules consist of a left hand side (*lhs* and a right hand side *rhs*. *lhs* of a rule has chunks and POS tags, whereas, *rhs* has reordered positions of those chunks and POS tags. Multiple rules can have same *lhs*. Rules can also consist of monotone ordered chunk sequences. Figure 5 shows some examples of the reordering rules.



no.	lhs	rhs
1.	$NP_0\ PP_1\ u_2\ n_3$	0 1 2 3
2.	$NP_0\ PP_1\ u_2\ n_3$	3 0 1 2
3.	$DNP_0\ NP_1\ VP_2$	0 1 2
4.	$DNP_0\ NP_1\ VP_2$	1 0 2
5.	$DNP_0\ NP_1\ m_2$	0 1 2
6.	$DNP_0\ NP_1\ m_2\ ad_3$	3 0 1 2
7.	$DNP_0\ NP_1\ m_2\ ad_3\ v_4$	4 3 0 1 2

Figure 5: Examples of reordering rules [Zhang et al., 2007]

Reordering rules are extracted based on the word alignments and source chunks. We get word alignments from GIZA++ aligner. We need to convert these word-to-word alignments into chunk-to-word alignments. Then, we apply phrase-extraction algorithm on these chunk-to-word alignments. We discard cross phrases. Figure 6(c) shows an example of a cross phrase. Phrases in Figure 6(a) and Figure 6(b) are accepted for reordering rules. Figure 6(a) is a monotone phrase, whereas Figure 6(b) is a reordering phrase.

Reordering lattice generation:

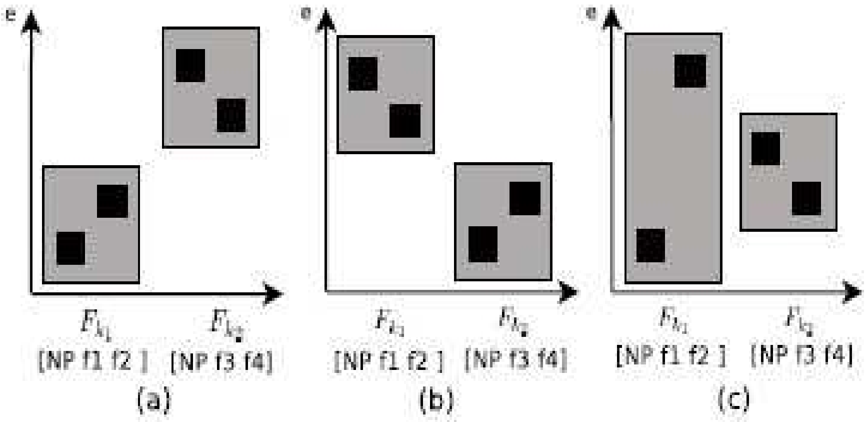
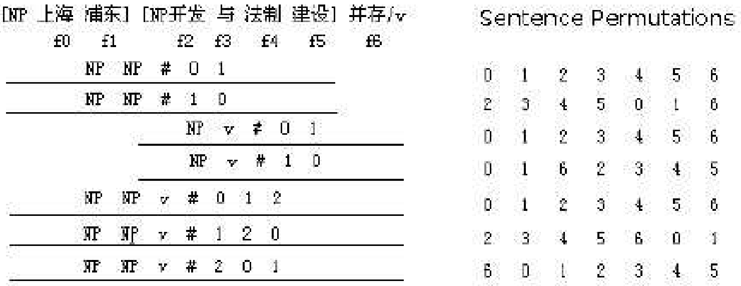


Figure 6: Examples of phrase extraction [Zhang et al., 2007]

After chunking the source sentence, we look for reordering rules of which *lhs* matches to any tag sequence of the input sentence. Thus, many paths are generated based on the rule applied. For the words uncovered by the rules, we use their POS tags. Figure 7 shows an example of application of reordering rules to a chinese sentence.



[NP 上海 浦东] [NP开发 与 法制 建设] 并存/v f0 f1 f2 f3 f4 f5 f6	Sentence Permutations
NP NP # 0 1	0 1 2 3 4 5 6
NP NP # 1 0	2 3 4 5 0 1 6
NP v # 0 1	0 1 2 3 4 5 6
NP v # 1 0	0 1 6 2 3 4 5
NP NP v # 0 1 2	0 1 2 3 4 5 6
NP NP v # 1 2 0	2 3 4 5 6 0 1
NP NP v # 2 0 1	6 0 1 2 3 4 5

Figure 7: Example of rule application [Zhang et al., 2007]

Each reordering S thus generated is stored in a lattice and given a weight W. Weight is computed using source language model ($p(S)$). Besides word N-gram model, a POS tag N-gram model or a chunk tag N-gram model can be used as well. These models can be learnt from tagged source training corpus.

Thus, we studied different approaches of reordering that can be used with statistical machine translation.

Summary

- We studied distance-based distortion reordering module and lexicalized reordering
- We studied limitations of distortion reordering module and studied source reordering
- We studied how parse tree-based source reordering model can be learnt for English
- We studied how chunk-based source reordering model can be learnt for Indian languages

References

Genzel, Dmitriy. Automatically learning source-side reordering rules for large scale machine translation. *Proceedings of the 23rd international conference on computational linguistics, Association for Computational Linguistics*, 2010.

Koehn, Philipp, Franz Josef Och, and Daniel Marcu. Statistical phrase-based translation. *Proceedings of the 2003 Conference of the North American Association for Computational Linguistics on Human Language Technology*, pages 48–54, 2003.

Koehn, Philipp and Hieu Hoang. Factored translation models. *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, ACL*, pages 868–876, 2007.

Koehn, Philipp. Statistical machine translation. *Cambridge University Press*, 2010

Patel, Raj Nath, Rohit Gupta, Prakash Pimpale, and Sasikumar M. Reordering rules for english-hindi smt. *Proceedings of the Second Workshop on Hybrid Approaches to Translation, Association for Computational Linguistics*, pages 34–41, 2013.

Ramanathan, Ananthakrishnan, Bhattacharyya P., Hegde J.J., Shah R.M., and Sasikumar M. Simple syntactic and morphological processing can help english-hindi statistical machine translation. *Proceedings of IJCNLP*, 2008.

Zhang, Yuqi, Richard Zens, and Hermann Ney. Chunk-level reordering of source language sentences with automatically learned rules for statistical machine translation. *Proceedings of the NAACL-HLT 2007/AMTA Workshop on Syntax and Structure in Statistical Translation, Association for Computational Linguistics*, 2007.