Supertag Based Pre-ordering in Machine Translation

Rajen Chatterjee, Anoop Kunchukuttan, Pushpak Bhattacharyya
Department of Computer Science and Engineering
Indian Institute of Technology, Bombay
`{rajen,anoopk,pb}@cse.iitb.ac.in`

Abstract

This paper presents a novel approach to integrate *mildly context sensitive grammar* in the context of pre-ordering for machine translation. We discuss the linguistic insights available in this grammar formalism and use it to develop a pre-ordering system. We show that mildly context sensitive grammar proves to be beneficial over context free grammar, which facilitates better reordering rules. For English to Hindi, we see significant improvement of 1.8 BLEU and error reduction of 4.46 TER score over CFG based pre-ordering system, on the WMT14 data set. We also show that our approach improves translation quality for English to Indian languages machine translation over standard phrase based systems.

1 Introduction

India is a multilingual country with 22 official languages, spanning four language families (Indo-Aryan,Dravidian, Tibeto-Burman and Austro-Asiatic)[1]. In such a liguistically diverse country there is always a great need for translation services to serve government, business and overall social communication needs. Hindi and English act as *link languages* across the country and languages of official communication for the Union Government. Thus, the importance of English to Hindi translation is obvious.

Languages can be differentiated in terms of structural divergences and morphological usage. English is structurally classified as a Subject-Verb-Object (SVO) language as opposed to Hindi which follows Subject-Object-Verb (SOV) word order. Largely, these divergences are responsible for the difficulties in standard statistical machine translation (SMT) systems.

Our objective is to reduce the structural divergence by reordering words in the source language (English) to conform to the target language (Hindi) word order and then provide this data to train a pharse based SMT system. This approach is known as *pre-ordering* in SMT research. The novelty of our work is the inclusion of linguistic context obtained from higher level of grammar formalism known as *mildly context sensitive grammar*. To the best of our knowledge, this is the first approach to bring such a formalism for pre-ordering in machine translation.

To begin with, Section 2 discusses work related to pre-ordering. We then provide an introduction to TAG and Supertag, in Section 3. Section 4 describes our approach for pre-ordering. In Section 5, we provide the methodology for pre-ordering. Experimental setup is explained in Section 6, while corresponding results are shown in Section 7. We conclude our work in Section 8 providing acknowledgement in Section 9.

2 Related Work

Word order has direct impact on the fluency of transaltion obtained using SMT systems. There are basically two paradigms for generating correct word order. The first paradigm deals with developing a reordering model which is used during decoding. Different solutions such as syntax-based models (Chiang, 2005), lexicalized reordering (Och et al., 2004), and tree-to-string methods (Zhang et al., 2006) have been proposed. Most of these approaches use statistical models to learn reordering rules, but all of them have different methods to solve the reordering problem. The next paradigm deals with developing a reordering model which is used as pre-processing step (also known as pre-ordering) in SMT systems. In pre-ordering, the objective is to re-

[1] `http://en.wikipedia.org/wiki/Languages_of_India`

order source language words to conform to the target language word order.

Xia and McCord (2004) describe an approach for translation from French to English, where reordering rules are acquired automatically using source and target parses and word alignment. The reordering rules in their approach operate at the level of context-free rules in the parse tree. Collins et al. (2005), describe clause restructuring for German to English machine translation. They use six transformations that are applied on German parsed text to reorder it before training with a phrase based system. Popovic and Ney (2006) use hand-made rules to reorder the source side based on POS information. Zhang et al. (2007) propose chunk level reordering, where reordering rules are automatically learned from source-side chunks and word alignments. They allow all possible reorderings to be used to create a lattice that is input to the decoder. Genzel (2010), shows automatic rule extraction for 8 language pairs. They first extract a dependency tree and then converts it to a shallow constituent tree. The trees are annotated by both POS tags and by Stanford dependency types, then they learn reordering rules given a set of features. This paper discusses about creating manual reordering rules with the help of Tree Adjoining Grammar (TAG), a mildly context sensitive formalism as discussed in Section 3.

3 Introduction to TAG/Supertag

Tree Adjoining Grammar (TAG) was introduced by Joshi et al. (1975). Tree Adjoining Languages (TALs) generate some strictly context-sensitive languages and fall in the class of the so-called *mildly context-sensitive* languages. Lexicalized Tree Adjoining Grammar (LTAG) (Joshi and Schabes, 1991) is the TAG formalism where each lexical item is associated with atleast one elementary structure. This elementary structure of LTAG is known as *Supertag*. The concept of *Supertag* was first proposed by Joshi and Srinivas (1994). Supertag localize dependencies, including long-distance dependencies, by requiring that all and only the dependent elements be present within the same structure. They provide syntactic as well as dependency information at the word level by imposing complex constraints in a local context. Supertags provide information like POS tag, subcategorization and other syntactic constraints at the level of agreement feature. Supertag can also be viewed as fragments of parse tree associated with each lexical item. An example of supertag is shown in Figure 1. This supertag is used for transitive verbs. Subcategorization information is

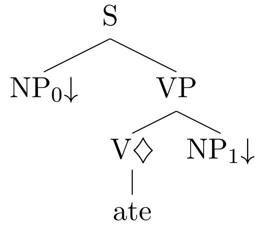


Figure 1: Supertag for transitive verb *ate*

clearly visible in the verb, which takes a subject to its left and an object to its right.

3.1 Structural description of supertag

Each supertag, at its frontier, has exactly one anchor node marked by $\diamondsuit$, to which a lexical item gets anchored. Apart from anchor node at its frontier, it may have an optional *substitution* node marked by $\downarrow$ or an *adjunction* node marked by *. *Substitution* and *adjunction* are those nodes which can be replaced by another supertag. For an adjunction node, it is necessary for its label to match the label of root node of supertag. A supertag can have atmost one adjunction node but can have more than one substitution node.

3.2 Supertagging

Supertagging refers to tagging each word of a sentence with a supertag. An example of a supertagged sentence is shown in figure 2.

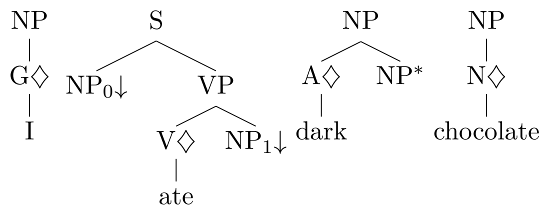


Figure 2: Supertagged sentence *"I ate dark chocolate"*

We use MICA Parser (Bangalore et al.,

2009)[2] to obtain rich linguistic information like POS tag, supertag, voice, the presence of empty subjects (PRO), wh-movement, deep syntactic argument (deep subject, deep direct object, deepindirect object), whether a verb heads a relative clause and also dependency relation, for each word. From this rich set of features, for each word, we extract word ID (essentially word position in a sentence), supertag, dependency relation and deep syntactic argument. Given a dependency relation, these supertags can be assembled using composition operation of TAG to form a constituent tree. Composition operation includes substitution and adjunction:

- **Substitution:** It deals with substituting a non-terminal node at the frontier of supertag with another supertag. An example of substitution operation is shown in Figure 3. When substitution occurs at a node, the node is replaced by the supertag to be substituted.

- **Adjunction:** It deals with inserting a supertag within another supertag satisfying adjunction constraints (Joshi, 1987). An example of adjoining operation is shown in Figure 4. One of the key contraint is the root label of supertag to be adjoined should match label of the node where adjunction occurs.

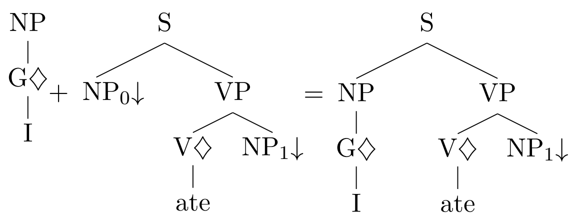


Figure 3: Example illustraing *Substitution* operation

4 Supertag Based Reordering

Each supertag encapsulates rich linguistic context, based on which, we reorder supertag nodes to preserve same context in target language. We first highlight linguistic relevance of supertag followed by showing why supertag facilitates reordering and finally illustrate reordering with supertag.

[2] http://mica.lif.univ-mrs.fr/

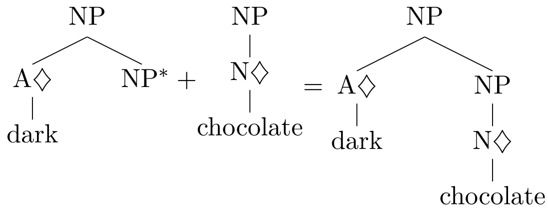


Figure 4: Example illustraing *Adjunction* operation

4.1 Linguistic Relevance

In this section, we show the linguistic relevance of supertag. Each Supertag falls in a specific syntactic environment, a few examples of supertag along with their syntactic environment are shown in Table 1. For each supertag ID, Table 1 gives the linguistic context (syntactic environment) in which a supertag appears along with an example sentence. The lexical item attached to the anchor node of supertag is written in *italic*, in the example sentence. Table 2 shows supertag ID and their corresponding tree structure.

Supertag ID	Linguistic Context	Example
t3	Noun Phrase	the *man*
t87	Topicalization	Those dogs, I am *terrified* of
t68	Imperative	Please *tidy* your room.
t36	Adjectival Modifier	*red* hat
t27	Transitive Verb	I *brought* dark chocolates.

Table 1: Example showing syntactic environment of different supertag

Syntactic environment of a supertag gives us the liberty to control reordering at a finer granularity. Thus, supertags of tree adjoining grammar prove to be advantageous over context free grammar (CFG) for reordering.

4.2 Why Supertag facilitates Reordering

In this section, we discuss key properties of LTAG which facilitates reordering, such as lexicalization, followed by Extended Domain of Locality (EDL) and finally Factoring of Recursion from the Domain of dependency (FRD)

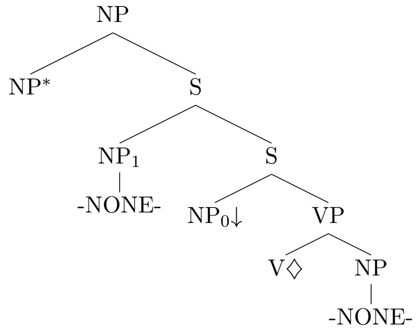
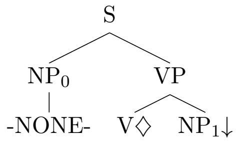
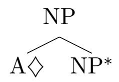
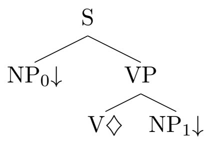
Supertag ID	Supertag Structure
t3	
t87	
t68	
t36	
t27	

Table 2: Figure showing supertag structure for each supertag ID

- **Lexicalization:**
  Lexicalization ensures that each supertag is anchored by a lexical item. This property prove to be linguistically crucial since it establishes a direct connection between the lexicon and the syntactic structures defined in the grammar. Reordering syntactic structure at word level for every word in a sentence helps us to obtain word order that conforms to target language word order.

- **Extended Domain of Locality (EDL):**
  This property of LTAG states that for each lexical item, the grammar must contain an elementary structure for each syntactic environment, the lexical item might appear in. This means that for each lexical item there can be multiple supertags representing different syntactic environment. Another part of EDL states that every supertag must contain all and only the argument of anchor in the same structure. This allows the anchor to impose syntactic and semantic constraints on its arguments directly since they appear in the same elementary structure that it anchors. This ensures, to reorder we deal only with primary features of lexical item on which the reordering primarily depends on.

- **Factoring of Recursion from the Domain of dependency (FRD):** This property ensures that supertags are *minimal* in nature; i.e. there is no recursion present within the given elementary structure of supertag. Recursive constructs are carried out with the help of adjunction operation. Auxiliary supertag, by adjunction to other supertag, accounts for long distance behavior of these dependencies. Due to this property the feature space of supertag is simplified without any recursive feature involved. This helps to build simple but strong reordering rules by just considering few but most important features of the supertag.

4.3 Detecting Linguistic Patterns for Reordering

In this section, we show that linguistic patterns can be detected with the help of supertag. English is structurally classified as a Subject-Verb-Object (SVO) language whereas Hindi is Subject-Object-Verb (SOV) language. However, this is not true for all verbs, specially in case of raising verb. We compare control verb v/s raising verb, where the former requires reordering as opposed to the later. An example has been shown in Table 3. It can be seen that supertag provides finer granularity to control reordering based on linguistic features. Studying supertag in-depth and associating it with linguistic context will help to discover various pattern useful for building better reordering system.

5 Methodology for Reordering

Listed below are the sequence of modules of our reordering system:

- **Reorder Supertags**
  We begin with reordering supertags, for

Control Verb		Raising Verb	
Sentence	Supertag	Sentence	Supertag
E: I <u>told</u> him. **H:** मैंने उसे <u>बोला</u> । **T:** mainne use <u>bola</u> **G:** I him <u>told</u>	S $NP_0\downarrow$ VP $V\diamond$ $NP_1\downarrow$	It <u>seems</u> you are busy. <u>लगता</u> है कि तुम व्यस्त हो । <u>lagta</u> hai ki tum vyast ho <u>seems</u> is that you busy are	S $NP_0\downarrow$ VP $V\diamond$ $S_1\downarrow$

Table 3: Table showing comparision of control verb v/s raising verb
E:English; H:Hindi; T:Transliteration; G:Gloss

which we are provided with 4725 supertags [3] in all, but seldom all of them are seen in action. Therefore, we supertag 50,000 sentences from health and tourism domain of ILCI corpus (Jha, 2010), from which we filter out spurious supertags which occur less than 10 times, leaving 600 supertags with us, for analysis. Our manual analysis shows that out of 600 supertags, 200 required reordering transformations to be applied.

Example of supertag, representing transitive verb, before and after reordering in shown in Table 4

Before Reordering	After Reordering
S $NP_0\downarrow$ VP $V\diamond$ $NP_1\downarrow$	S $NP_0\downarrow$ VP $NP_1\downarrow$ $V\diamond$

Table 4: Example showing transitive verb before and after reordering

An example of reordering prepositional phrase which gets adjunct to a noun phrase is shown in Table 5.

Before Reordering	After Reordering
NP NP^* PP $IN\diamond$ $NP_1\downarrow$	NP PP NP^* $NP_1\downarrow$ $IN\diamond$

Table 5: Example of prepositional phrase before and after reordering

[3] http://mica.lif.univ-mrs.fr/

- **Extract Linguistic Information**
  As stated in Section 3.2, we use MICA Parser to obtain linguistic information along with superatg at word level. For example, consider the input sentence “I told him”, for which we extract essential information and append it with respective words as shown “I|1|2|PRP|t29|0 told|2|2|VBD|t27|. him|3|2|PRP|t29|1”. Here, each word contains “lexical-item|ID|parent-ID|POStag|Supertag|deep-argument-position”. We carry this example sentence in further modules of reordering.

- **Construct Derivation Tree**
  Once linguistic information is extracted, we proceed with construction of derivation tree. The process of combining the elementary trees (supertags) to yield a parse of the sentence is represented by the derivation tree. Derivation tree for the example sentence is shown in Figure 5.

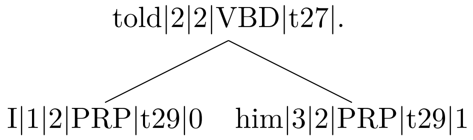


Figure 5: Example of Derivation Tree

- **Construct Derived Tree**
  We have implemented the composition operation (adjunction and substitution), which, given a derivation tree would build a derived/parse tree. Derived tree for the example sentence is shown in Figure 6. To build this derived tree we use reordered version of supertag (for eg. t27 for transitive verb from Table 4).

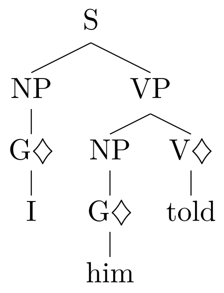


Figure 6: Example of Derived Tree

- **Extract leaf nodes** Finally, we extract leaf nodes from the derived tree, which gives us reordered English sentence. In our case, the input sentence is "I told him" for which, we get "I him told " as output. We now use these reordered English sentences along with their parallel Hindi translation to train English to Hindi machine translation system.

6 Experiment Setup

In this section, we discuss different data set used and various experiments that we have performed.

6.1 Data Sets

We use the shared task data set provided in *Workshop on Statistical Machine Translation, 2014* (WMT14)[4] to train English to Hindi translation system. WMT14 training set contains data from general domain, obtained from multiple sources, whereas test set belongs to news domain. The training set consist of 275K parallel segments, while test set contains 2.5K. We train a 5 gram language model using SRILM[5] on 1.5M monolingual Hindi corpus provided in WMT14 shared task.

For translation from English to multiple Indian languages we use *Indian Language Corpora Initiative* (ILCI) (Jha, 2010) corpus. ILCI data belongs to the health and tourism domain. The training set consist of 40K parallel sentences while test set contains 1.1K. We train a 5 gram language model using SRILM on a 50K monolingual Hindi corpus from the same domain.

The English data was tokenized using the Stanford tokenizer and then true-cased using truecase.perl provided in MOSES toolkit. We normalize Hindi corpus using NLP Indic Library (Kunchukuttan et. al.,2014)[6]. Normalization is followed by tokenization, wherein we make use of the trivtokenizer.pl provided with WMT14 shared task.

6.2 List of Experiments

We use the MOSES toolkit (Koehn et al., 2007) to train various statistical machine translation systems. For English to Hindi we train three systems, on each data set (WMT14 and ILCI), as follows:

- **Phrase Based Systems:** We train phrase based model without pre-ordering.

- **Context Free Grammar based preordering:** In this model, we reorder both train and test set using Patel et al. (2013)'s source side reordering rules which is refinement of Ramanathan et al. (2008)'s rule-based reordering system.

- **Supertag based pre-ordering:** In this model, we reorder both train and test set using our supertag based pre-ordering method as discussed in Section 5.

We provide systematic comparision among these systems in Section 7. As the reordering rules are developed to conform Hindi word order, we were interested to see how does it affect other Indian languages which have the same word order as Hindi. So, we developed various translation systems from English to other Indian languages using ILCI corpus and compare it with standard pharse based systems.

7 Results

In this section, we also provide quantitative results for various systems and provide systematic comparision among them. We also provide couple of translations from English to Hindi, showing improvement in translation quality with our approach.

[4] WMT14 resources:-`http://ufallab.ms.mff.cuni.cz/~bojar/hindencorp/`

[5] `http://www.speech.sri.com/projects/srilm`

[6] `https://bitbucket.org/anoopk/indic_nlp_library`

Data Set	Phrase Based		CFG based reordering		Supertag based reordering	
	BLEU	TER	BLEU	TER	BLEU	TER
WMT14	8.00	84.00	8.6	86.66	**10.40**	**82.20**
ILCI	23.77	58.36	**26.45**	56.24	25.75	**56.14**

Table 6: Result of English to Hindi SMT System
Higher BLEU Score and Lower TER Score indicate better system

Language Pair	Baseline Phrase Based		Supertag based reordering		Improvement over Baseline (BLEU %)
	BLEU	TER	BLEU	TER	
English-Punjabi	20.71	62.94	22.60	60.64	+9.12
English-Urdu	16.20	69.12	17.43	67.38	+7.59
English-Konkani	10.18	79.35	11.18	78.83	+9.82
English-Telugu	4.8	95.41	5.44	95.46	+13.33
English-Gujarati	14.76	70.42	15.40	69.34	+4.33
English-Bengali	12.73	75.49	13.26	74.48	+4.16
English-Malayalam	3.49	100.23	3.78	100.39	+8.31

Table 7: Result of English to Indian Language SMT System

7.1 Quantitative Evaluation

We use two standard evaluation metrics BLEU (Papineni et al., 2002) and TER (Snover et al., 2006), for comparing translation quality of various systems. Table 6 compares phrase based, CFG based reordering and supertag based reordering systems built using WMT14 and ILCI data set, for English to Hindi language pair. We see TER score of our system is best for both data sets. For WMT14 data set, our approach shows improvement of 1.8 BLEU score and error reduction of 4.46 TER score over CFG based reordering system.

For the ILCI data set, our approach shows significant improvement over both evaluation metrics over phrase based SMT sytem but shows slight degradation in BLEU score when compared with CFG based reordering system. However, TER score remains almost same. On an average, over both data sets, we see that TAG based reordering performs better than PB and CFG based systems.

Also from Table 7, we see our approch of reordering English data set proves to give significant improvement in translation quality over phrase based, for most of the English to Indian language machine translation systems. The result is statistically significant at the p $<=$ 0.07 level.

7.2 Qualitative Evaluation

We provide two source (English) sentences (Example 1 and Example 2) along with their actual translation (AT) and machine translations obtained from phrase based system (PB), CFG based reordered system (CFG) and TAG based reordered system (TAG), as shown in Table 8. Transliteration of all Hindi sentences have been shown in corresponding column. We see that translations of TAG is much closer to the actual translation. Example 1 shows improvement in reordering which is the main focus of our work, whereas, Example 2 shows morphological improvement inherently achieved with reordering.

8 Conclusion

We presented a novel method of using mildly context sensitive grammar formalism in the context of pre-ordering in SMT systems. We show that the rich linguistic information embedded in supertags provides finer granularity for framing reordering rules over Context Free Grammar (CFG). We also showed that supertag can be used for detecting linguistic pattern based on which specific reordering rules can be written. We also discuss the linguistic relevance of Supertag and bridging it with machine translation. Finally our approach has shown significant improvement over CFG based reordered systems. In future

Example 1	One may wear clothes in many folds on body .	
AT	शरीर पर कई तह में कपड़े पहनें ।	sharir par kai taha mein kapade pahane .
PB	सकती है पर कई तह में कपड़े पहनें ।	sakti hai par kai taha mein kapade pahane .
CFG	एक कई तह में कपड़े शरीर पर पहन सकते हैं ।	ek kai taha mein kapade sharir par pahan sakte hain .
STAG	एक शरीर पर कई तह में कपड़े पहनें ।	ek sharir par kai taha mein kapde pahane .
Example 2	The symptoms of which are as follows :	
AT	जिसके लक्षण इस प्रकार हैं :	jisake lakshan is prakar hain :
PB	जो के लक्षण इस प्रकार हैं :	jo ke lakshan is prakar hain :
CFG	जो के लक्षण इस प्रकार हैं :	jo ke lakshan is prakar hain :
STAG	जिसके लक्षण इस प्रकार हैं :	jisake lakshan is prakar hain :

Table 8: Comparision of translation among various systems for English to Hindi

work, we would like to classify supertag based on their linguistic relevance and try to generalize reordering rules for each class.

9 Acknowledgement

We thank Owen Rambow and Srinivas Bangalore to help us in understanding supertags and also making the grammar file of supertags available to us. We would like to thank the Technology Development for Indian Languages (TDIL) Programme and the Department of Electronics Information Technology, Govt. of India for providing the ILCI corpus. Also we would like to thank all members of the Centre for Indian Language Technology (CFILT) for their valuable feedback and comments.

References

Srinivas Bangalore, Pierre Boulllier, Alexis Nasr, Owen Rambow, and Benoît Sagot. 2009. Mica: a probabilistic dependency parser based on tree insertion grammars application note. In *Proceedings of HLT-NAACL*, pages 185–188.

David Chiang. 2005. A hierarchical phrase-based model for statistical machine translation. In *Proceedings of the 43rd Annual Meeting on ACL*, pages 263–270.

Michael Collins, Philipp Koehn, and Ivona Kučerová. 2005. Clause restructuring for statistical machine translation. In *Proceedings of the ACL*, pages 531–540.

Dmitriy Genzel. 2010. Automatically learning source-side reordering rules for large scale machine translation. In *Proceedings of the COLING*, pages 376–384.

Girish Nath Jha. 2010. The tdil program and the indian language corpora initiative (ilci). In *Proceedings of the LREC*.

Aravind K Joshi and Yves Schabes. 1991. Tree-adjoining grammars and lexicalized grammars.

Aravind K Joshi and Bangalore Srinivas. 1994. Disambiguation of super parts of speech (or supertags): Almost parsing. In *Proceedings of the COLING*, pages 154–160.

Aravind K Joshi, Leon S Levy, and Masako Takahashi. 1975. Tree adjunct grammars. *Journal of computer and system sciences*, 10(1):136–163.

Aravind K Joshi. 1987. An introduction to tree adjoining grammars. *Mathematics of language*, 1:87–115.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the ACL*, pages 177–180.

Franz Josef Och, Daniel Gildea, Sanjeev Khudanpur, Anoop Sarkar, Kenji Yamada, Alexander Fraser, Shankar Kumar, Libin Shen, David Smith, Katherine Eng, et al. 2004. A smorgasbord of features for statistical machine translation. In *Proceedings of the HLT-NAACL*, pages 161–168.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the ACL*, pages 311–318.

Raj Patel, Rohit Gupta, Prakash Pimpale, and M. Sasikumar. 2013. Reordering rules for English-Hindi SMT. In *Proceedings of the Second Workshop on Hybrid Approaches to Translation*.

Maja Popovic and Hermann Ney. 2006. Pos-based word reorderings for statistical machine translation. In *Proceedings of the LREC*, pages 1278–1283.

Ananthakrishnan Ramanathan, Jayprasad Hegde, Ritesh M Shah, Pushpak Bhattacharyya, and M Sasikumar. 2008. Simple syntactic and morphological processing can help english-hindi statistical machine translation. In *Proceedings of the IJCNLP*, pages 513–520.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of the AMTA*, pages 223–231.

Fei Xia and Michael McCord. 2004. Improving a statistical mt system with automatically learned rewrite patterns. In *Proceedings of the COLING*, page 508.

Hao Zhang, Liang Huang, Daniel Gildea, and Kevin Knight. 2006. Synchronous binarization for machine translation. In *Proceedings of the HLT-NAACL*, pages 256–263.

Yuqi Zhang, Richard Zens, and Hermann Ney. 2007. Chunk-level reordering of source language sentences with automatically learned rules for statistical machine translation. In *Proceedings of the HLT-NAACL*, pages 1–8.