A Discriminative Approach for Dependency Based Statistical Machine Translation

Sriram Venkatapathy
LTRC, IIIT-Hyderabad
sriram@research.iiit.ac.in

Rajeev Sangal
LTRC, IIIT-Hyderabad
sangal@mail.iiit.ac.in

Aravind Joshi
University of Pennsylvania
joshi@seas.upenn.edu

Karthik Gali[1]
Talentica
karthik.gali@gmail.com

Abstract

In this paper, we propose a dependency based statistical system that uses discriminative techniques to train its parameters. We conducted experiments on an English-Hindi parallel corpora. The use of syntax (dependency tree) allows us to address the large word-reorderings between English and Hindi. And, discriminative training allows us to use rich feature sets, including linguistic features that are useful in the machine translation task. We present results of the experimental implementation of the system in this paper.

1 Introduction

Syntax based approaches for Machine Translation (MT) have gained popularity in recent times because of their ability to handle long distance reorderings (Wu, 1997; Yamada and Knight, 2002; Quirk et al., 2005; Chiang, 2005), especially for divergent language pairs such as English-Hindi (or English-Urdu). Languages such as Hindi are also known for their rich morphology and long distance agreement of features of syntactically related units. The morphological richness can be handled by employing techniques that factor the lexical items into morphological factors. This strategy is also useful in the context of English-Hindi MT (Bharati et al., 1997; Bharati et al., 2002; Ananthakrishnan et al., 2008; Ramanathan et al., 2009) where there is very limited parallel corpora available, and breaking words into smaller units helps in reducing sparsity. In order to handle phenomenon such as long-distance word agreement to achieve accurate generation of target language words, the inter-dependence between the factors of syntactically related words need to be modelled effectively.

Some of the limitations with the syntax based approaches such as (Yamada and Knight, 2002; Quirk et al., 2005; Chiang, 2005) are, (1) They do not offer flexibility for adding linguistically motivated features, and (2) It is not possible to use morphological factors in the syntax based approaches. In a recent work (Shen et al., 2009), linguistic and contextual information was effectively used in the framework of a hierarchical machine translation system. In their work, four linguistic and contextual features are used for accurate selection of translation rules. In our approach in contrast, linguistically motivated features can be defined that directly effect the prediction of various elements in the target during the translation process. This features use syntactic labels and collocation statistics in order to allow effective training of the model.

Some of the other approaches related to our model are the Direct Translation Model 2 (DTM2) (Ittycheriah and Roukos, 2007), End-to-End Discriminative Approach to MT (Liang et al., 2006) and Factored Translation Models (Koehn and Hoang, 2007). In DTM2, a discriminative trans-

[1]This work was done at LTRC, IIIT-Hyderabad, when he was a masters student, till July 2008

Proceedings of SSST-4, Fourth Workshop on Syntax and Structure in Statistical Translation, pages 66–74,
COLING 2010, Beijing, August 2010.

lation model is defined in the setting of a phrase based translation system. In their approach, the features are optimized globally. In contrast to their approach, we define a discriminative model for translation in the setting of a syntax based machine translation system. This allows us to use both the power of a syntax based approach, as well as, the power of a large feature space during translation. In our approach, the weights are optimized in order to achieve an accurate prediction of the individual target nodes, and their relative positions.

We propose an approach for syntax based statistical machine translation which models the following aspects of language divergence effectively.

- *Word-order variation* including long-distance reordering which is prevalent between language pairs such as English-Hindi and English-Japanese.
- *Generation of word-forms* in the target language by predicting the word and its factors. During prediction, the *inter-dependence of factors* of the target word form with the factors of syntactically related words is considered.

To accomplish this goal, we visualize the problem of MT as transformation from a morphologically analyzed source syntactic structure to a target syntactic structure[1] (See Figure 1). The transformation is factorized into a series of mini-transformations, which we address as features of the transformation. The features denote the various linguistic modifications in the source structure to obtain the target syntactic structure. Some of the examples of features are lexical translation of a particular source node, the ordering at a particular source node etc. These features can be entirely local to a particular node in the syntactic structure or can span across syntactically related entities. More about the features (or mini-transformations) is explained in section 3. The transformation of a source syntactic structure is scored by taking a weighted sum of its features [2]. Let τ represent the transformation of source syntactic structure s, the score of transformation is computed as represented in Equation 1.

$$score(\tau|s) = \sum_i w_i * f_i(\tau, s) \qquad (1)$$

In Equation 1, $f_i's$ are the various features of transformation and $w_i's$ are the weights of the features. The strength of our approach lies in the flexibility it offers in incorporating linguistic features that are useful in the task of machine translation. These features are also known as *prediction features* as they map from source language information to information in the target language that is being predicted.

During decoding a source sentence, the goal is to choose a transformation that has the highest score. The source syntactic structure is traversed in a bottom-up fashion and the target syntactic structure is simultaneously built. We used a bottom-up traversal while decoding because it builds a contiguous sequence of nodes for the subtrees during traversal enabling the application of a wide variety of language models.

In the training phase, the task is to learn the weights of features. We use an online large-margin training algorithm, MIRA (Crammer et al., 2005), for learning the weights. The weights are locally updated at every source node during the bottom-up traversal of the source structure. For training the translation model, automatically obtained word-aligned parallel corpus is used. We used GIZA++ (Och and Ney, 2003) along with the *growing* heuristics to word-align the training corpus.

The basic factors of the word used in our experiments are root, part-of-speech, gender, number and person. In Hindi, common nouns and verbs have gender information whereas, English doesn't contain that information. Apart from the basic factors, we also consider the role information provided by labelled dependency parsers. For computing the dependency tree on the source side, We used stanford parser (Klein and Manning, 2003) in the experiments presented in this chapter[3].

[1]Note that target structure contains only the target factors. An accurate and deterministic morphological generator combines these factors to produce the target word form.

[2]The features can be either binary-values or real-valued

[3]Stanford parser gives both the phrase-structure tree as well as dependency relations for a sentence.

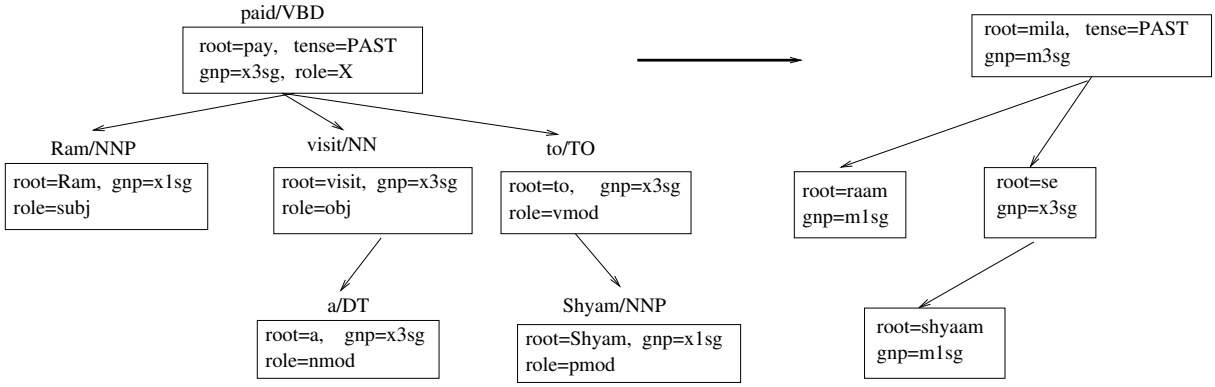

Figure 1: Transformation from source structure to target language

The function words such as prepositions and auxiliary verbs largely express the grammatical roles/functions of the content words in the sentence. In fact, in many agglutinative languages, these words are commonly attached to the content word to form one word form. In this paper, we also conduct experiments where we begin by grouping the function words with their corresponding function words. These groups of words are called local-word groups. In these cases, the function words are considered as factors of the content words. Section 2 explains more about the local word groups in English and Hindi.

2 Local Word Groups

Local word groups (LWGs) (Bharati et al., 1998; Vaidya et al., 2009) consist of a content word and its associated function words. Local word grouping reduces a sentence to a sequence of content words with the case-markers and tense-markers acting as their factors. For example, consider an English sentence 'People of these island have adopted Hindi as a means of communication'. 'have adopted' is a LWG with root 'adopt' and tense markers being 'have_ed'. Another example for the LWG will be 'of communication' where 'communication' is the root, and 'of' is the case-marker. It is to be noted that Local word grouping is different from chunking, where more than one content word can be part of a chunk. We obtain local word groups in English by processing the output of the stanford parser. In Hindi, the function words always appear immediately after the content word[4], and it requires simple pattern

matching to obtain the LWGs. The rules applied are, (1) VM (RB|VAUX)+, and (2) N.* IN.

3 Features

There are three types of transformation features explored by us, (1) Local Features, (2) Syntactic Features and, (3) Contextual Features. In this section, we describe each of these categories of features representing different aspects of transformation with examples.

3.1 Local Features

The local features capture aspects of local transformation of an *atomic treelet* in the source structure to an atomic treelet in the target language. Atomic treelet is a semantically non-decomposible group of one or more nodes in the syntactic structure. It usually contains only one node, except for the case of multi-word expressions (MWEs). Figure 2 presents the examples of local transformation.

Some of the local features used by us in our experiments are (1) *dice coefficient*, (2) dice coefficient of roots, (3) dice coefficient of null translations, (4) *treelet translation probability*, (5) *gnp-gnp pair*, (5) *preposition-postposition pair*, (6) *tense-tense pair*, (7) *part-of-speech fertility* etc. *Dice coefficients* and *treelet translation probabilities* are measures that express the statistical co-occurrence of the atomic treelets.

[4]case-markers are called postpositions

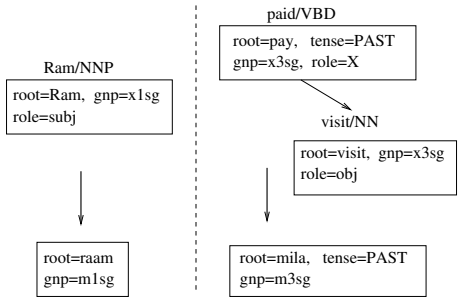

Figure 2: Local transformations

3.2 Syntactic Features

The syntactic features are used to model the difference in the word orders of the two languages. At every node of the source syntactic structure, these features define the changes in the relative order of children during the process of transformation. They heavily use source information such as part-of-speech tags and syntactic roles of the source nodes. One of the features used is *reorderPostags*.

This feature captures the change in relative positions of children with respect to their parents during the tree transformation. An example feature for the transformation given in Figure 1 is shown in Figure 3.

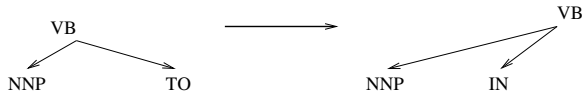

Figure 3: Syntactic feature - reorder postags

The feature *reorderPostags* is in the form of a complete transfer rule. To handle cases, where the left-hand side of 'reorderPostags' does not match the syntactic structure of the source tree, the simpler feature functions are used to qualify various reorderings. Instead of using POS tags, feature functions can be defined that use syntactic roles.

Apart from the above feature functions, we can also have features that compute the score of a particular order of children using syntactic language models (Gali and Venkatapathy, 2009; Guo et al., 2008). Different features can be defined that use different levels of information pertaining to the atomic treelet and its children.

3.3 Contextual Features

Contextual features model the inter-dependence of factors of nodes connected by dependency arcs. These features are used to enable access to global information for prediction of target nodes (words and its factors).

One of the features *diceCoeffParent*, relates the parent of a source node to the corresponding target node (see figure 4.

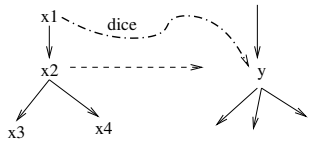

Figure 4: Use of Contextual (parent) information of $x2$ for generation of y

The use of this feature is expected to address of the limitations of using 'atomic treelets' as the basic units in contrast to phrase based systems which consider arbitrary sequences of words as units to encode the local contextual information. In my case, We relate the target treelet with the contextual information of the source treelet using feature functions rather than using larger units. Similar features are used to connect the context of a source node to the target node.

Various feature functions are defined to handle interaction between the factors of syntactically related treelets. The gender-number-person agreement is a factor that is dependent of gender-number-person factors of the syntactically related treelets in Hindi. The rules being learnt here are simple. However, more complex interactions can also be handled though features such as *prep_Tense* where, the case-marker in the target is linked to the tense of parent verb.

4 Decoding

The goal is to compute the most probable target sentence given a source sentence. First, the source sentence is analyzed using a morphological analyzer[5], local word grouper (see section 2) and a dependency parser. Given the source structure, the task of the decoding algorithm is to choose the transformation that has the maximum score.

[5]http://www.cis.upenn.edu/∼xtag/

The dependency tree of the source language sentence is traversed in a bottom-up fashion for building the target language structure. At every source node during the traversal, the local transformation is first computed. Then, the relative order of its children is then computed using the syntactic features. This results in a target structure associated with the subtree rooted at the particular node. The target structure associated with the root node of the source structure is the result of the best transformation of the entire source structure.

Hence, the task of computing the best transformation of the entire source structure is factorized into the tasks of computing the best transformations of the source treelets. The equation for computing the score of a transformation, Equation 1, can be modified as Equation 2 given below.

$$score(\tau|s) = \sum_{r} |r| * \sum_{i} w_i * f_i(\tau_r, r) \quad (2)$$

where, τ_j is the local transformation of the source treelet r. The best transformation $\hat{\tau}$ of source sentence s is,

$$\hat{\tau} \quad = \quad argmax_{\tau}\ score(\tau|s) \quad (3)$$

5 Training Algorithm

The goal of the training algorithm is to learn the feature weights from the word aligned corpus. For word-alignment, we used the IBM Model 5 implemented in GIZA++ along with the *growing* heuristics (Koehn et al., 2003). The gold atomic treelets in the source and their transformation is obtained by mapping the source node to the target using the word-alignment information. This information is stored in the form of transformation tables that is used for the prediction of target atomic treelets, prepositions and other factors. The transformation tables are pruned in order to limit the search and eliminate redundant information. For each source element, only the top few entries are retained in the table. This limit ranges from 3 to 20.

We used an online-large margin algorithm, MIRA (McDonald and Pereira, 2006; Crammer et al., 2005), for updating the weights. During parameter optimization, it is sometimes impossible to achieve the gold transformation for a node because the pruned transformation tables may not lead to the target gold prediction for the source node. In such cases where the gold transformation is unreachable, the weights are not updated at all for the source node as it might cause erroneous weight updates. We conducted our experiments by considering both the cases, (1) Identifying source nodes with unreachable transformations, and (2) Updating weights for all the source nodes (till a maximum iteration limit). The number of iterations on the entire corpus can also be fixed. Typically, two iterations have been found to be sufficient to train the model.

The dependency tree is traversed in a bottom-up fashion and the weights are updated at each source node.

6 Experiments and Results

The important aspects of the translation model proposed in this paper have been implemented. Some of the components that handle word insertions and non-projective transformations have not yet been implemented in the decoder, and should be considered beyond the scope of this paper. The focus of this work has been to build a working syntax based statistical machine translation system, which can act as a platform for further experiments on similar lines. The system would be available for download at http://shakti.iiit.ac.in/∼sriram/vaanee.html. To evaluate this experimental system, a restricted set of experiments are conducted. The experiments are conducted on the English-Hindi language pair using a corpus in tourism domain containing 11300 sentence pairs[6].

6.1 Training

6.1.1 Configuration

For training, we used DIT-TOURISM-ALIGN-TRAIN dataset which is the word-aligned dataset of 11300 sentence pairs. The word-alignment is done using GIZA++ (Och and Ney, 2003) toolkit and then *growing* heuristics are applied. For our experiments, we use two *growing* heuristics, GROW-DIAG-FINAL-AND and GROW-DIAG-FINAL as they cover most number of words in both the sides of the parallel corpora.

[6]DIT-TOURISM corpus

Number of Training Sentences	500
Iterations on Corpus	1-2
Parameter optimization algorithm	MIRA
Beam Size	1-20
Maximum update attempts at source node	1-4
Unreachable updates	False
Size of transformation tables	3

Table 1: Training Configuration

The training of the model can be performed under different configurations. The configurations that we used for the training experiments are given in Table 6.1.1.

6.2 Results

For the complete training, the number of sentences that should be used for the best performance of the decoder should be the complete set. In the paper, we have conducted experiments by considering 500 training sentences to observe the best training configuration.

At a source node, the weight vector is iteratively updated till the system predicts the gold transformation. We conducted experiments by fixing the maximum number of update attempts. A source node, where the gold transformation is not achieved even after the maximum updates limit, the update at this source node is termed a *update failure*. The source nodes, where the gold transformation is achieved even without making any updates is known as the *correct prediction*.

At some of the source nodes, it is not possible to arrive at the gold target transformation because of limited size of the training corpus. At such nodes, we have avoided doing any weight update. As the desired transformation is unachievable, any attempt to update the weight vector would cause noisy weight updates.

We observe various parameters to check the effectiveness of the training configuration. One of the parameters (which we refer to as 'updateHits') computes the number of successful updates (S) performed at the source nodes in contrast to number of failed updates (F). Successful updates result in the prediction of the transformation that is same as the reference transformation. A failed update doesn't result in the achievement of the correct prediction even after the maximum iteration limit (see section 6.1.1) is reached. At some of the source nodes, the reference transformations are *unreachable* (U). The goal is to choose the configuration that has least number of average failed updates (F) because it implies that the model has been learnt effectively.

			UpdateHit			
	K	**m**	P	S	F	U
1.	1	4	1680	2692	84	4081
2.	5	4	1595	2786	75	4081
3.	10	4	1608	2799	49	4081
4.	20	4	1610	2799	**47**	4081

Table 2: Training Statistics - Effect of Beam Size

From Table 2, we can see that the bigger beam size leads to a better training of the model. The beam size was varied between 1 and 20, and the number of update failures (F) was observed to be least at K=20.

			UpdateHit			
	K	**m**	P	S	F	U
1.	20	1	1574	2724	158	4081
2.	20	2	1598	2767	91	4081
3.	20	4	1610	2799	**47**	4081

Table 3: Training Statistics - Effect of maximum update attempts

In Table 3, we can see that an higher limit on the maximum number of update attempts results in less number of update attempts as expected. A much higher value of m is not preferable because the training updates makes noisy updates in case of *difficult* nodes i.e., the nodes where target transformation is reachable in theory, but is unreachable given the set of features.

			UpdateHit			
	K	**i**	P	S	F	U
1.	1	1	1680	2692	84	4081
2.	1	2	1679	2694	83	4081

Table 4: Training Statistics - Effect of number of iterations

Now, we examine the effect of number of it-

erations on the quality of the model. In table 4, we can observe that the number of iterations on the data has no effect on the quality of the model. This implies, that the model is adequately learnt after one pass through the data. This is possible because of the multiple number of update attempts allowed at every node. Hence, the weights are updated at a node till the model prediction is consistent with the gold transformation.

Based on the above observations, we consider the configuration 4 in Table 2 for the decoding experiments.

Now, we present some of the top features weights leant by the best configuration. The weights convey that important properties of transformation are being learnt well. Table 5 presents the weights of the features 'diceRoot', 'diceRootChildren' and 'diceRootParent'.

Feature	Weight
dice	75.67
diceChildren	540.31
diceParent	595.94
treelet translation probability (ttp) 1	0.77
treelet translation probability (ttp) 2	389.62

Table 5: Weights of dice coefficient based features

We see that the dice coefficient based local and contextual features have a positive impact on the selection of correct transformations. A feature that uses a syntactic language model to compute the perplexity per word has a negative weight of **-1.115**.

Table 6 presents the top-5 entries of contextual features that describe the translation of source argument 'nsubj' using contextual information ('tense' of its parent).

Feature	Weight
roleTenseVib:nsubj+NULL___NULL	44.194196513246
roleTenseVib:nsubj+has_VBN___ne	14.4541356715382
roleTenseVib:nsubj+VBD___ne	10.9241093097953
roleTenseVib:nsubj+VBP___meM	6.14149937079584
roleTenseVib:nsubj+VBP___NULL	4.76795730621754

Table 6: Top weights of a contextual feature : preposition+Tense-postposition

Table 7 presents the top-10 ordering relative position feature where the head word is a verb. In this feature, the relative position (left or right) of the head and the child is captured. For example, a feature 'relPos:amod-NN', if active, conveys that an argument with the role 'amod' is at the left of a head word with POS tag 'NN'.

Feature	Weight
relPos:amod-NN	6.70
relPos:NN-appos	1.62
relPos:lrb-NN	1.62

Table 7: Top weights of *relPos* feature

6.3 Decoding

We computed the translation accuracies using two metrics, (1) BLEU score (Papineni et al., 2002), and (2) Lexical Accuracy (or F-Score) on a test set of 30 sentences. We compared the accuracy of the experimental system (Vaanee) presented in this paper, with Moses (state-of-the-art translation system) and Shakti (rule-based translation system [7]) under similar conditions (with using a development set to tune the models). The rule-based system considered is a general domain system tuned to the tourism domain. The best BLEU score for Moses on the test set is **0.118**, and the best lexical accuracy is **0.512**. The best BLEU score for Shakti is **0.054**, and the best lexical accuracy is **0.369**.

In comparison, the best BLEU score of Vaanee is **0.067**, while the best lexical accuracy is **0.445**. As observed, the decoding results of the experimental system mentioned here are not yet comparable to the state-of-art. The main reasons for the low translation accuracies are,

1. Poor Quality of the dataset

   The dataset currently available for English-Hindi language pair is noisy. This is an extremely large limiting factor for a model which uses rich linguistic information within the statistical framework.

2. Low Parser accuracy

[7] http://shakti.iiit.ac.in/

The parser accuracy on the English-Hindi dataset is low, the reasons being, (1) Noise, (2) Length of sentences, and (3) Wide scope of the tourism domain.

3. Word insertions not implemented yet
4. Non-projectivity not yet handled
5. BLEU is not an appropriate metric

   BLEU is not an appropriate metric (Ananthakrishnan et al.,) for measuring the translation accuracy into Indian languages.

6. Model is context free as far as targets words are concerned. Selection depends on children but not parents and siblings

   This point concerns the decoding algorithm. The current algorithm is greedy while chosing the best translation at every source node. It first explores the K-best local transformations at a source node. It then makes a greedy selection of the predicted subtree based on it's overall score after considering the predictions at the child nodes, and the relative position of the local transformation with respect the predictions at the child nodes.

   The problem in this approach is that, an error once made at a lower level of the tree is propogated to the top, causing more mistakes. A computationally reasonable solution to this problem is to maintain a K-best list of predicted subtrees corresponding to every source node. This allows rectification of a mistake made at any stage.

The system, however, performs better than the rule based system. As observed earlier, the right type of information is being learnt by the model, and the approach looks promising. The limitations expressed here shall be addressed in the future.

7 Conclusion

In this work, we presented a syntax based dependency model to effectively handle problems in translation from English to Indian languages such as, (1) Large word order variation, and (2) Accurate generation of word-forms in the target language by predicted the word and its factors. The model that we have proposed, has the flexibility of adding rich linguistic features.

An experimental version of the system has been implemented, which is available for download at http://shakti.iiit.ac.in/∼sriram/vaanee.html. This can facilitate as a platform for future research in syntax based statistical machine translation from English to Indian languages. We also plan to perform experiments using this system between European languages in future.

The performance of the implemented translation system, is not yet comparable to the state-of-art results primarily for two reasons, (1) Poor quality of available data, because of which our model which uses rich linguistic information doesn't perform as expected, and (2) Components for word insertion and non-projectivity handling are yet to be implemented in this version of the system.

References

Ananthakrishnan, R, B Pushpak, M Sasikumar, and Ritesh Shah. Some issues in automatic evaluation of english-hindi mt: more blues for bleu. ICON-2007.

Ananthakrishnan, R., Jayprasad Hegde, Pushpak Bhattacharyya, and M. Sasikumar. 2008. Simple syntactic and morphological processing can help english-hindi statistical machine translation. In *Proceedings of IJCNLP-2008*. IJCNLP.

Bharati, Akshar, Vineet Chaitanya, Amba P Kulkarni, and Rajeev Sangal. 1997. Anusaaraka: Machine translation in stages. *A Quarterly in Artificial Intelligence, NCST, Bombay (renamed as CDAC, Mumbai)*.

Bharati, Akshar, Medhavi Bhatia, Vineet Chaitanya, and Rajeev Sangal. 1998. Paninian grammar framework applied to english. *South Asian Language Review*, (3).

Bharati, Akshar, Rajeev Sangal, Dipti M Sharma, and Amba P Kulkarni. 2002. Machine translation activities in india: A survey. In *Proceedings of workshop on survey on Research and Development of Machine Translation in Asian Countries*.

Chiang, David. 2005. A hierarchical phrase-based model for statistical machine translation. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 263–270, Ann Arbor, Michigan, June. Association for Computational Linguistics.

Crammer, K., R. McDonald, and F. Pereira. 2005. Scalable large-margin online learning for structured classification. Technical report, University of Pennsylvania.

Gali, Karthik and Sriram Venkatapathy. 2009. Sentence realisation from bag of words with dependency constraints. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Student Research Workshop and Doctoral Consortium*, pages 19–24, Boulder, Colorado, June. Association for Computational Linguistics.

Guo, Yuqing, Josef van Genabith, and Haifeng Wang. 2008. Dependency-based n-gram models for general purpose sentence realisation. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 297–304, Manchester, UK, August. Coling 2008 Organizing Committee.

Ittycheriah, Abraham and Salim Roukos. 2007. Direct translation model 2. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 57–64, Rochester, New York, April. Association for Computational Linguistics.

Klein, Dan and Christopher D. Manning. 2003. Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 423–430, Sapporo, Japan, July. Association for Computational Linguistics.

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

Koehn, P., F. J. Och, and D. Marcu. 2003. Statistical phrase-based translation. In *Proceedings of the Human Language Technology Conference 2003 (HLT-NAACL 2003)*, Edmonton, Canada, May.

Liang, P., A. Bouchard-Cote, D. Klein, and B. Taskar. 2006. An end-to-end discriminative approach to machine translation. In *International Conference on Computational Linguistics and Association for Computational Linguistics (COLING/ACL)*.

McDonald, R. and F. Pereira. 2006. Online learning of approximate dependency parsing algorithms. In *EACL*.

Och, F.J. and H. Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

Papineni, Kishore, Salim Roukos, Todd Ward, and W.J. Zhu. 2002. Bleu: A method for automatic evaluation of machine translation. In *Proceedings of 40^{th} Annual Meeting of the Association of Computational Linguistics*, pages 313–318, Philadelphia, PA, July.

Quirk, Chris, Arul Menezes, and Colin Cherry. 2005. Dependency treelet translation: Syntactically informed phrasal SMT. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 271–279, Ann Arbor, Michigan, June. Association for Computational Linguistics.

Ramanathan, Ananthakrishnan, Hansraj Choudhary, Avishek Ghosh, and Pushpak Bhattacharyya. 2009. Case markers and morphology: Addressing the crux of the fluency problem in english-hindi smt. In *Proceedings of ACL-IJCNLP 2009*. ACL-IJCNLP.

Shen, Libin, Jinxi Xu, Bing Zhang, Spyros Matsoukas, and Ralph Weischedel. 2009. Effective use of linguistic and contextual information for statistical machine translation. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 72–80, Singapore, August. Association for Computational Linguistics.

Vaidya, Ashwini, Samar Husain, Prashanth Reddy, and Dipti M Sharma. 2009. A karaka based annotation scheme for english. In *Proceedings of CICLing , 2009*.

Wu, Dekai. 1997. Stochastic Inversion Transduction Grammars and Bilingual Parsing of Parallel Corpora. *Computational Linguistics*, 23(3):377–404.

Yamada, Kenji and Kevin Knight. 2002. A decoder for syntax-based statistical mt. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, pages 303–310, Philadelphia, Pennsylvania, USA, July. Association for Computational Linguistics.