A Dependency Based Statistical Translation Model

Giuseppe Attardi
Università di Pisa
Dipartimento di Informatica
attardi@di.unipi.it

Atanas Chanev
Università di Pisa
Dipartimento di Informatica
chanev@di.unipi.it

Antonio Valerio Miceli Barone
Università di Pisa
Dipartimento di Informatica
miceli@di.unipi.it

Abstract

We present a translation model based on dependency trees. The model adopts a tree-to-string approach and extends Phrase-Based translation (PBT) by using the dependency tree of the source sentence for selecting translation options and for reordering them. Decoding is done by translating each node in the tree and combining its translations with those of its head in alternative orders with respect to its siblings. Reordering of the siblings exploits a heuristic based on the syntactic information from the parse tree which is learned from the corpus. The decoder uses the same phrase tables produced by a PBT system for looking up translations of single words or of partial sub-trees. A mathematical model is presented and experimental results are discussed.

1 Introduction

Several efforts are being made to incorporate syntactic analysis into phrase-base statistical translation (PBT) (Och 2002; Koehn et. al. 2003), which represents the state of the art in terms of robustness in modeling local word reordering and efficiency in decoding. Syntactic analysis is meant to improve some of the pitfalls of PBT:

- Translation options selection: candidate phrases for translation are selected as consecutive n-grams. This may miss to consider certain syntactic phrases if their component words are far apart.
- Phrase reordering: especially for languages with different word order, e.g. subject-verb-object (SVO) and subject-object-verb (SVO) languages, long distance reordering is a problem. This has been addressed with a distance based distortion model (Och 2002; Koehn et al. 2003), lexicalized phrase reordering (Tillmann, 2004; Koehn, et.al., 2005; Al-Onaizan and Papineni, 2006), by hierarchical phrase reordering model (Galley and Manning, 2008) or by reordering the nodes in a dependency tree (Xu et al., 2009)
- Movement of translations of fertile words: a word with fertility higher than one can be translated into several words that do not occur consecutively. For example, the Italian sentence "*Lui partirà domani*" translates into German as "*Er wird morgen abreisen*". The Italian word "*partirà*" (meaning "*will leave*") translates into "*wird gehen*" in German, but the infinite "*abreisen*" goes to the end of the sentence with a movement that might be quite long.

Reordering of phrases is necessary because of different word order typologies of languages: constituent word order like SOV for Hindi vs. SVO for English; order of modifiers like noun–adjective for French, Italian vs. adjective-noun in English. Xu et al. (2009) tackle this issue by introducing a reordering approach based on manual rules that are applied to the parse tree produced by a dependency parser.

However the splitting phenomenon mentioned above requires more elaborate solutions than simple reordering grammatical rules.

Several schemes have been proposed for improving PBMT systems based on dependency trees. Our approach extends basic PBT as de-

Proceedings of SSST-5, Fifth Workshop on Syntax, Semantics and Structure in Statistical Translation, pages 79–87,
ACL HLT 2011, Portland, Oregon, USA, June 2011.

scribed in (Koehn et. al., 2003) with the following differences:

- we perform tree-to-string translation. The dependency tree of the source language sentence allows identifying syntactically meaningful phrases as translation options, instead of n-grams. However these phrases are then still looked up in a Phrase Translation Table (PT) quite similarly to PBT. Thus we avoid the sparseness problem that other methods based on treelets suffer (Quirk et al., 2005).
- reordering of phrases is carried out traversing the dependency tree and selecting as options phrases that are children of each head. Hence a far away but logically connected portion of a phrase can be included in the reordering.
- phrase combination is performed by combining the translations of a node with those of its head. Hence only phrases that have a syntactic relation are connected. The Language Model (LM) is still consulted to ensure that the combination is proper, and the overall score of each translation is carried along.
- when all the links in the parse tree have been reduced, the root node contains candidate translations for the whole sentences
- alternative visit orderings of the tree may produce different translations so the final translation is the one with the highest score.

Some of the benefits of our approach include:

1) reordering is based on syntactic phrases rather than arbitrary chunks
2) computing the future cost estimation can be avoided, since the risk of choosing an easier n-gram is mitigated by the fact that phrases are chosen according to the dependency tree
3) since we are translating from tree to string, we can directly exploit the standard phrase tables produced by PBT tools such as giza++ (Och and Ney, 2000) and Moses (Koehn, 2007)
4) integration with the parser: decoding can be performed incrementally while a dependency Shift/Reduce parser builds the parse tree (Attardi, 2006).

2 The Dependency Based Decoder

We describe in more detail the approach by presenting a simple example.

The translation of an input sentence is generated by reducing the dependency tree one link at a time, i.e. merging one node with its parent and combining their translations, until a single node remains. Links must be chosen in an order that preserves the connectivity of the dependency tree. Since there is a one-to-one correspondence between links and nodes (i.e. the link between a node and its head), we can use any ordering that corresponds to a *topological ordering* of the nodes of the tree.

A sentence is a sequence of words (w_1, … , w_n), so we can use their index to identify words and hence each ordering is a permutation of those indexes.

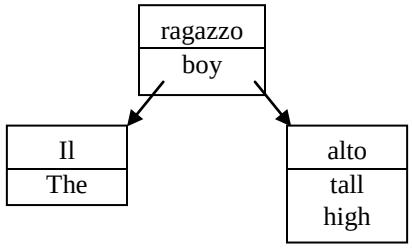
Consider for example the dependency tree for the Italian sentence: *Il ragazzo alto* ("*The tall boy*").

There are only two possible topological orderings for this tree: 1-3-2 and 3-1-2.

In principle the decoding process should explore all possible topological orderings for generating translations, but their number is too big, being proportional to the factorial of the number of words, so we will introduce later a criterion for selecting a subset of these, which conform best with the rules of the languages.

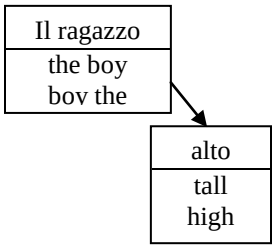
Given a permutation we obtain a translation by merging in that order each node with its parent.

The initialization step of the decoder creates nodes corresponding to the parse tree and collects translations for each individual word from the PT.



Case 1: Permutation 1-3-2

The first merge step is applied to the nodes for w_1 and its head w_2, performing the concatenation of the translations of nodes *il* (*the*) and *ragazzo* (*boy*), both in normal and reverse order. Hence expansion of this hypothesis reduces the tree to the following, where we show also the partial translations associated to each node. Each translation has associated weights (i.e. the LM weight, the translation model weight, etc.) and a cumulative score. The score is the dot product of the weights for the sentence and the vector of tuning parameters for the model. The score is used to rank the sentences and also to limit how many of them are kept according to the beam size parameter of the algorithm.



The second step merges the node for word w_3 ("*alto*") with that of its head w_2 ("*ragazzo*") producing a single node with four translations: "*the boy tall*", "*boy the tall*", "*tall the boy*" and "*tall boy the*".

Il ragazzo alto
the boy tall
boy the tall
tall the boy
tall boy the

Case 2: Permutation 3-1-2

The first merge between w_3 and w_2 generates two translation fragments: "*boy tall*" and "*tall boy*". The second one creates four translations: "*the boy tall*", "*boy tall the*", "*the tall boy*", "*tall boy the*".

When the tree has been reduced to a single root node and the results of both permutations are collected, the node will contain all eight alternative translations ranked according to the language model, so that the best one, possibly "*the tall boy*", can be selected as overall sentence translation.

3 Node Merge

The operation of node merge consists of taking all possible translations for the two nodes and concatenating them in either sequential or reverse order, adding them to the translation of the parent node and dropping the child.

In certain cases though, for example idiomatic phrases, the best translation is not obtained by combining the individual translations of each word, but instead a proper translation might be found in the Phrase Translation Table (PT). Hence besides performing combination of translations, we also consider the sub-tree rooted at the head node h_{ri} of node r_i. We consider the phrase corresponding to the leaves of the sub-tree rooted at h_{ri} and all children already merged into it, including r_i: if this phrase is present in the PT, then its translations are also added to the node.

This is sometimes useful, since it allows the decoder to exploit phrases that only correspond to partial sub-trees that it will otherwise miss.

4 Reordering Rules

In order to restrict the number of permutations to consider, we introduce a reordering step based on rules that examine the dependency tree of the source sentence.

The rules are dependent on the language pair and they can be learned automatically from the corpus.

We report first a simple set of hand crafted rules devised for the pair Italian-English that we used as a baseline.

The default ordering is to start numbering the left children of a node backwards, i.e. the node closer to the head comes first, then continuing with the right children in sequential order.

Special rules handle these cases:

1) The head is a verb: move an adverb child to first position. This lets a sequence of VA VM V R be turned into VA VM R V, where VA is the POS for auxiliary verbs, VM for modals, V for main verb and R for adverbs.
2) The head is a noun: move adjectives or prepositions immediately following the head to the beginning.

4.1 Learning Reordering Rules

In order to learn the reordering rules we created a word-aligned parallel corpus from 1.3 million source sentences selected from the parallel corpus. The corpus is parsed and each parse tree is analyzed using the giza++ word alignments of its translation to figure out node movements.

For each source-language word, we estimate a unique alignment to a target-language word. If the source word is aligned to more than one target word we select the first one appearing in the alignment file. If a source word is not aligned to any word, we choose the first alignment in its descendants in the dependency tree. If no alignment can be found in the descendants, we assume that the word stays in its original position.

We reorder the source sentence according to this alignment, putting it in target-language order.

We produce a training event consisting of a pair (*context, offset*) for each non-root word. The context of the event consists of a set of features (the POS tag of a word, its dependency tag and the POS of its head) extracted for the word and its children. The outcome of the event is the offset of the word relative to its parent (negative for words that appear on the left of their parent in target-language order, positive otherwise).

We calculate the relative frequency of each event conditioned on the *context*, deriving rules of the form:

(*context, offset, Pr*[*Offset = offset | Context = context*]).

During decoding, we compute a reordering position for each source word by adding to the word position to the offset predicted by the most likely reordering rule matching the word context (or 0 if no matching context is found).

The reordering position drives the children combination procedure in the decoder.

Our reordering rules are similar to those proposed by Xu at al. (2009), except that we derive them automatically from the training set, rather than being hand-coded.

4.2 Beam Search

Search through the space of hypotheses generated is performed using beam search that keeps in each node the list of the top best translations for the node. The score for the translation is computed using the weights of the individual phrases that make up the translation and the overall LM probability of the combination.

The scores are computed querying the standard Moses Phrase Table and the LM for the target language; other weights uses by moses such as the reordering weights or the future cost estimates are discarded or not computed.

5 The Model

A mathematical model of the dependency based translation process can be formulated as follows.

Consider the parse of a sentence f of length n. Let R denote all topological ordering of the nodes according to the dependency tree.

Let f_r denote the parse tree along with a consistent node ordering r. Each ordering gives rise to several different translations. Let E_r denote the set of translations corresponding to f_r. We assign to each translation $e_r \in E_r$ a probability according to the formula below. The final translation is the best result obtained through combinations over all orderings.

Error! Objects cannot be created from editing field codes.

Where e_r denotes any of the translations of f obtained when nodes are combined according to node ordering r.

The probability of a translation e_r corresponding to a node ordering r for a phrase f, $p(e_r \mid f)$ is defined as:

Error! Objects cannot be created from editing field codes.

where

Error! Objects cannot be created from editing field codes. and**Error! Objects cannot be created from editing field codes.**denote the leaf words from node r_i and those of its head node h_{ri}, respectively.

Error! Objects cannot be created from editing field codes.is either **Error! Objects cannot**

be created from editing field codes.or **Error! Objects cannot be created from editing field codes.**

$p(f, e) = p_{PT}(str(f), e)$ if $str(f) \in PT$

$str(f)$ is the sentence at the leaves of node r_i

p_{LM} is the Language Model probability

p_{PT} is the Phrase Table probability

6 Related Work

Yamada and Knight (2001) introduced a syntax-based translation model that incorporated source-language syntactic knowledge within statistical translation. Many similar approaches are based on constituent grammars, among which we mention (Chiang, 2005) who introduced hierarchical translation models.

The earliest approach based on dependency grammars is the work by Ashlawi et al. (2000), who developed a tree-to-tree translation model, based on middle-out string transduction capable of phrase reordering. It translated transcribed spoken utterances from English to Spanish and from English to Japanese. Improvements were reported over a word-for-word baseline.

Ambati (2008) presents a survey of other approaches based on dependency trees.

Quirk et. al. (2005) explore a *tree-to-tree* approach, called treelet translation, that extracts treelets, i.e. sub-trees, from both source and target language by means of a dependency parser. A word aligner is used to align the parallel corpus. The source dependency is projected onto the target language sentence in order to extract treelet translation pairs. Given a foreign input sentence, their system first generates its dependency tree made of treelets. These treelets are translated into treelets of the target language, according to the dependency treelet translation model. Translated treelets are then reordered according to a reorder model.

The ordering model is trained on the parallel corpus. Treelet translation pairs are used for decoding. The reordering is done at the treelet level where all the child nodes of a node are allowed all possible orders. The results show marginal improvements in the BLEU score (40.66) in comparison with Pharaoh and MSR-MT. But the treelet translation algorithm is more than an order of magnitude slower.

Shen et. al. (2008) present a hierarchical machine translation method from *string to trees*. The scheme uses the dependency structure of the target language to use transfer rules while generating a translation. The scheme uses well-formed dependency structure which involves fixed and floating type structures. The floating structures allow the translation scheme to perform different concatenation, adjoining and unification operations still being within the definition of well-formed structures. While decoding the scheme uses the probability of a word being the root, and also the left-side, right-side generative probabilities. The number of rules used varies from 27 M (for a string to dependency system) to 140 M (baseline system). The performance reached 37.25% for the system with 3-grams, 39.47% for 5-grams.

Marcu and Wong (2002) propose a joint- probability model. The model establishes a correspondence between a source phrase and a target phrase through some concept. The reordering is integrated into the joint probability model with the help of:

3) Phrase translation probabilities **Error! Objects cannot be created from editing field codes.** denoting the probability that concept c_i generates the translation **Error! Objects cannot be created from editing field codes.** for the English and **Error! Objects cannot be created from editing field codes.** for the foreign language inputs.
4) Distortion probabilities based on absolute positions of the phrases.

Decoding uses a hill-climbing algorithm. Performance wise the approach records an average BLEU score of 23.25%, with about 2% of improvement over the baseline IBM system.

Zhang et. al. (2007) present a reordering model that uses linguistic knowledge to guide both phrase reordering and translation between linguistically correct phrases by means of rules. Rules are encoded in the form of weighted synchronous grammar and express transformations on the parse trees. They experiment also mixing constituency and dependency trees achieving some improve-

ments in BLEU score (27.37%) over a baseline system (26.16%).

Cherry (2008) introduces a cohesion feature into a traditional phrase based decoder. It is implemented as a soft constraint which is based on the dependency syntax of the source language. He reports a BLEU score improvement on French-English translation.

The work by Xu et al. (2009) is the closest to our approach. They perform preprocessing of the foreign sentences by parsing them with a dependency parser and applying a set of hand written rules to reorder the children of certain nodes. The preprocessing is applied to both the training corpus and to the sentences to translate, hence after reordering a regular hierarchical system can be applied. Translation experiments between English and five non SVO Asian languages show significant improvements in accuracy in 4 out of 5 languages. With respect to our approach the solution by Xu et al. does not require any intervention on the translation tools, since the sentences are rewritten before being passed to the processing chain: on the other hand the whole collection has to undergo full parsing with higher performance costs and higher dependency on the accuracy of the parser.

Dyer and Resnik (2010) introduce a translation model based on a Synchronous Context Free Grammar (SCFG). In their model, translation examples are stored as a context-free forest. The process of translation comprise two steps: tree-based reordering and phrase transduction. While reordering is modeled with the context-free forest, the reordered source is transduced into the target language by a Finite State Transducer (FST). The implemented model is trained on those portions of the data which it is able to generate. An increase of BLEU score is achieved for Chinese-English when compared to the phrase based baseline.

Our approach is a true *tree-to-string* model and differs from (Xu et al., 2009), which uses trees only as an intermediate representation to rearrange the original sentences. We perform parsing and reordering only on the phrases to be translated. The training collection is kept in the original form, and this has two benefits: training is not subject to parsing errors and our system can share the same model of a regular hierarchical system.

Another difference is in the selection of translation options: our method exploits the parse tree to select grammatical phrases as translation options.

7 Implementation

The prototype decoder consists of the following components:

1) A specialized table lookup server, providing an XML-RPC interface for querying both the phrase table and the LM
2) A parser engine based on DeSR (DeSR, 2009)
3) A reordering algorithm that adds ordering numbers to the output produced by DeSR in CoNLL-X format. Before reordering, this step also performs a restructuring of the parse tree, converting from the conventions of the Italian Tanl Treebank to a structure that helps the analysis. In particular it converts conjunctions, which are represented as chains, where each conjunct connects to the previous, to a tree where they are all dependent of the same head word. Compound verbs are also revised: in the dependency tree each auxiliary of a verb is a direct child of the main verb. For example in "*avrebbe potuto vedere*", both the auxiliary "*avrebbe*" and the modal "*potuto*" depend on the verb "*vedere*". This steps groups all auxiliaries of a verb under the first one, i.e. "*potuto*". This helps so that the full auxiliary can be looked up separately from the verb in the phrase table.
4) A decoder that uses the output produced by the reordering algorithm, queries the phrase table and performs a beam search on the hypotheses produced according to the suggested reordering.

8 Experimental Setup and Results

Moses (Koehn et al., 2007) is used as a baseline phrase-based SMT system. The following tools and data were used in our experiments:

1) the IRSTLM toolkit (Marcello and Cettolo, 2007) is used to train a 5-gram language mod-

el with *Kneser-Ney* smoothing on a set of 4.5 million sentences from the Italian Wikipedia.

2) the Europarl version 6 corpus, consisting of 1,703,886 sentence pairs, is used for training. A tuning set of 2000 sentences from ACL WMT 2007 is used to tune the parameters.
3) the model is trained with lexical reordering.
4) the model is tuned with mert (Bertoldi, et al.)
5) the official test set from ACL WMT 2008 (Callison-Burch et al., 2008), consisting of 2000 sentences, is used as test set.
6) the open-source parser DeSR (DeSR, 2009) is used to parse Italian sentences, trained on the Evalita 2009 corpus (Bosco et al., 2009). Parser domain adaptation is obtained by adding to this corpus a set of 1200 sentences from the ACL WMT 2005 test set, parsed by DeSR and then corrected by hand.

Both the training corpora and the test set had to be cleaned in order to normalize tokens: for example the English versions contained possessives split like this "*Florence' s*". We applied the same tokenizer used by the parser which conforms to the PTB standard.

DeSR achieved a Labeled Accuracy Score of 88.67% at Evalita 2009, but for the purpose of translation, just the Unlabeled Accuracy is relevant, which was 92.72%.

The table below shows the results of our decoder (Desrt) in the translation from Italian to English, compared to a baseline Moses system trained on the same corpora and to the online version of Google translate.

Desrt was run with a beam size of 10, since experiments showed no improvements with a larger beam size.

We show two versions of Desrt, one with parse trees as obtained by the parser and one (Desrt gold) where the trees were corrected by hand. The difference is minor and this confirms that the decoder is robust and not much affected by parsing errors.

System	*BLEU*	*NIST*
Moses	**29.43**	**7.22**
Moses tree phrases	28.55	7.10
Desrt gold	26.26	6.88
Desrt	26.08	6.86
Google Translate	24.96	6.86
Desrt learned	24.37	6.76

Table 1. Results of the experiments.

Since we used the same phrase table produced by Moses also for Desrt, Moses has an advantage, because it can look up n-grams that do not correspond to grammatical phrases, which Desrt never considers. In order to determine how this affects the results, we tested Moses restricting its choice to phrases corresponding to treelets form the parse tree. The result is shown in the row in the table labeled as "Moses tree phrases". The score is lower, as expected, but this confirms that Desrt makes quite good use of the portion of the phrase table it uses.

Since the version of the reordering algorithm we used produces a single reordering, the Desrt decoder has linear complexity on the length of the sentence. Indeed, despite being written in Python and having to query the PT as a network service, it is quite faster than Moses.

9 Error Analysis

Despite that fact that Desrt is driven by the parse tree, it is capable of selecting fairly good and even long sentences for look up in the phrase table.

How close is the Desrt translation from those of the Moses baseline can be seen from this table:

	1-gram	2-gram	3-gram	4-gram	5-gram
NIST	7.28	3.05	1.0	0.27	0.09
BLEU	84.73	67.69	56.94	48.59	41.78

Sometimes Desrt fails to select a better translation for a verb, since it looks up prepositional phrases separately from the verb, while Moses often connects the preposition to the verb.

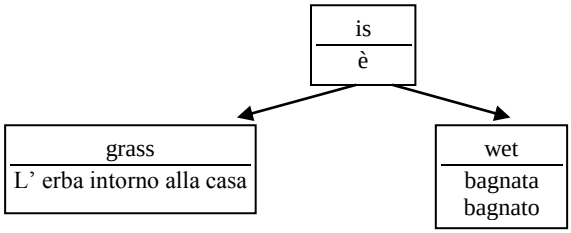
This could be improved by performing a check and scoring higher translations which include the translation of the preposition dependent on the verb.

Another improvement could come from creating phrase tables limited to treelet phrases, i.e. phrases corresponding to treelets from the parser.

10 Enhancements

The current algorithm needs to be improved to fully deal with certain aspects of long distance dependencies. Consider for example the sentence "*The grass around the house is wet*". The dependency tree of the sentence contains the non-contiguous phrases "*The grass*" and "*wet*", whose Italian translation must obey a morphological gender agreement between the subject "*grass*" ("*erba*", feminine), and the adjective "*wet*" ("*bagnata*").

However, the current combination algorithm does not exploit this dependence, because the last phases of node merge will occur when the tree has been reduced to this:



The PT however could tell us that "*erba bagnata*" is more likely than "*erba bagnato*" and allow us to score the former higher.

11 Conclusions

We have described a decoding algorithm guided by the dependency tree of the source sentence. By exploiting the dependency tree and deterministic reordering rules among the children of a node, the decoder is fast and can be kept simple by avoiding to consider multiple reorderings, to use reordering weights and to estimate future costs.

There is still potential for improving the algorithm exploiting information implicit in the PT in terms of morphological constraints, while maintaining a simple decoding algorithm that does not involve complex grammatical transformation rules.

The experiments show encouraging results with respect to state of the art PBT systems. We plan to test the system on other language pairs to see how it generalizes to other situations where phrase reordering is relevant.

Acknowledgments

Zauhrul Islam helped setting up our baseline system and Niladri Chatterjie participated in the early design of the model.

References

G. Attardi. 2006. Experiments with a Multilanguage Non-Projective Dependency Parser. *Proc. of the Tenth Conference on Natural Language Learning*, New York, (NY).

H. Alshawi, S. Douglas and S. Bangalore. 2000. Learning Dependency Translation Models as Collections of Finite State Head Transducers. *Computational Linguistics* 26(1), 45–60.

N. Bertoldi, B. Haddow, J-B. Fouet. 2009. Improved Minimum Error Rate Training in Moses. In *Proc. of 3rd MT Marathon*, Prague, Czech Republic.

V. Ambati. 2008. Dependency Structure Trees in Syntax Based Machine Translation. Adv. MT Seminar Course Report.

C. Bosco, S. Montemagni, A. Mazzei, V. Lombardo, F. Dell'Orletta and A. Lenci. 2009. Evalita'09 Parsing Task: comparing dependency parsers and treebanks. *Proc. of Evalita 2009*.

P. F. Brown, V. J. Della Pietra, S. A. and R. L. Mercer. 1993. The Mathematics of Statistical Machine Translation: Parameter Estimation. *Computational Linguistics*, 19(2), 263–311.

Callison-Burch et al. 2008. Further Meta-Evaluation of Machine Translation. *Proc. of ACL WMT 2008*.

C. Cherry. 2008. Cohesive phrase-based decoding for statistical machine translation. *Proc. of ACL 2008: HLT.*

D. Chiang. 2005. A hierarchical phrase-based model for statistical machine translation. In *Proc. of ACL 2005*.

DeSR. Dependency Shift Reduce parser. http://sourceforge.net/projects/desr/

Y. Ding, and M. Palmer. 2005. Machine Translation using Probabilistic Synchronous Dependency Insertion Grammar. *Proc. of ACL'05*, 541–548.

C. Dyer and P. Resnik. 2010. Context-free reordering, finite-state translation. *Proc. of HLT: The 2010 Annual Conference of the North American Chapter of the ACL*, 858–866.

F. Marcello, M. Cettolo. 2007. Efficient Handling of N-gram Language Models for Statistical Machine Translation. *Workshop on Statistical Machine Translation 2007*.

M. Galley and C. D. Manning. 2008. A Simple and Effective Hierarchical Phrase Reordering Model. In *Proc. of EMNLP 2008*.

R. Hwa, P. Resnik, A. Weinberg, C. Cabezas and O. Kolak, 2005. Bootstrapping Parsers via Syntactic Projection across Parallel texts. *Natural Language Engineering* 11(3), 311-325.

P. Koehn, F. J. Och and D. Marcu. 2003. Statistical Phrase-Based Translation. *Proc. of Human Language Technology and North American Association for Computational Linguistics Conference* (HLT/NAACL), 127–133.

P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proc. of the 45th Annual Meeting of the ACL*, demonstration session, 177–180, Prague, Czech Republic.

P. Koehn. 2009. *Statistical Machine Translation*. Cambridge University Press.

Y. Liu, Q. Liu and S. Lin. 2006. Tree-to-string Alignment Template for Statistical Machine Translation, In *Proc. of COLING-ACL*.

D. Marcu and W. Wong. 2002. A Phrase-Based Joint Probability Model for Statistical Machine Translation. *Proc. Empirical Methods in Natural Language Processing* (EMNLP), 133–139.

C. Quirk, A. Menzes and C. Cherry. 2005. Dependency Treelet Translation: Syntactically Informed Phrasal SMT. *Proc. 43rd Annual Meeting of the ACL*, 217–279.

S. Libin, J. Xu and R. Weischedel. 2008. A New String-to-Dependency Machine Translation Algorithm with a Target Dependency Language Model. *Proc. ACL-08*, 577–585.

F. J. Och 2002. Statistical Machine Translation: From Single Word Models to Alignment Template. Ph.D. Thesis, RWTH Aachen, Germany.

F.J. Och, H. Ney. 2000. Improved Statistical Alignment Models. *Proc. of the 38th Annual Meeting of the ACL*. Hong Kong, China. 440-447.

K. Yamada and K. Knight. 2001. A Syntax-Based Statistical Translation Model. *Proc. 39th Annual Meeting of ACL* (ACL-01), 6–11.

P. Xu, J. Kang, M. Ringgaard and F. Och. 2009. Using a Dependency Parser to Improve SMT for Subject-Object-Verb Languages. *Proc. of NAACL 2009*, 245–253, Boulder, Colorado.

D. Zhang, Mu Li, Chi-Ho Li and M. Zhou. 2007. Phrase Reordering Model Integrating Syntactic Knowledge for SMT. *Proc. Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Processing*: 533–540.