Advancements in Arabic-to-English Hierarchical Machine Translation

Matthias Huck[1] and **David Vilar**[1,2] and **Daniel Stein**[1] and **Hermann Ney**[1]

[1] Human Language Technology and Pattern
Recognition Group, RWTH Aachen University
<surname>@cs.rwth-aachen.de

[2] DFKI GmbH
Berlin, Germany
david.vilar@dfki.de

Abstract

In this paper we study several advanced techniques and models for Arabic-to-English statistical machine translation. We examine how the challenges imposed by this particular language pair and translation direction can be successfully tackled within the framework of hierarchical phrase-based translation.

We extend the state-of-the-art with a novel cross-system and cross-paradigm lightly-supervised training approach. In addition, for following recently developed techniques we provide a concise review, an empirical evaluation, and an in-depth analysis: soft syntactic labels, a discriminative word lexicon model, additional reorderings, and shallow rules. We thus bring together complementary methods that previously have only been investigated in isolation and mostly on different language pairs.

Combinations of the methods yield significant improvements over a baseline using a usual set of models. The resulting hierarchical systems perform competitive on the large-scale NIST Arabic-to-English translation task.

1 Introduction

Since its introduction in (Chiang, 2005), hierarchical phrase-based translation has become a standard approach in statistical machine translation. Many additional features and enhancements to the hierarchical paradigm have been proposed or adopted from the conventional phrase-based approach, but the effect of the various methods is typically merely evaluated separately. Neither are they compared to each other, nor is it clear whether combining the methods would be beneficial.

The aim of the work presented in this paper is to explore the effectiveness of a state-of-the-art hierarchical phrase-based system for large-scale Arabic-to-English statistical machine translation (SMT). Within this framework, we investigate the impact of several recently developed methods on the translation performance. Not only do we analyze them separately, but also examine whether their combination further increases the output quality.

More specifically, we focus on three models: First, we integrate syntactic information in order to improve the linguistic structure of the translation. Second, we utilize a discriminatively trained extended word lexicon to obtain a better lexical selection based on global source sentence context. Third, we introduce a jump model which is based on reordering enhancements to the hierarchical grammar to allow for more flexibility during the search process.

The Arabic-English language pair is known to behave more monotone than other language pairs, e.g. Urdu-English or Chinese-English. In a contrastive experiment done by Birch et al. (2009), a hierarchical system does not outperform a conventional phrase-based system for Arabic-English. On the other hand, a lattice-based hierarchical system (de Gispert et al., 2010) has been the best-performing system at the 2009 NIST Arabic-English evaluation campaign.[1] Noticing these

[1] http://www.itl.nist.gov/iad/mig/tests/mt/2009/ResultsRelease/currentArabic.html

facts, we also want to investigate to what extent the translation quality relies on the recursion depth for hierarchical rules. In order to separate the effect of the recursion level, we conduct all experiments with an unrestricted hierarchical grammar as well as with a depth-restricted one.

Finally, we perform a novel cross-system and cross-paradigm variant of lightly-supervised training (Schwenk, 2008). We make use of bitexts that have been built by automatic translation of large amounts of monolingual data with a conventional phrase-based system to improve our translation model. We propose to integrate this kind of data as purely lexicalized rules solely while sticking to the set of hierarchical rules that is extracted from the more reliable human-generated parallel data.

2 Overview

The paper is structured as follows: First we give an outline of some previous work that is related to ours (Section 3). We then present the methods we apply in the following sections:

We introduce *soft syntactic labels* in Section 4, an approach to integrate syntactic information in a non-obtrusive manner into hierarchical search as an additional model. The discriminatively trained extended word lexicon model that is employed in this work is discussed in Section 5. Section 6 contains a description of the reordering enhancement we apply to the hierarchical phrase-based model. In Section 7 we describe the limitation of the recursion depth for hierarchical rules. Section 8 presents an effective and easily implementable way to integrate information extracted from unsupervised training data into the translation model of a hierarchical phrase-based system.

We present the experimental setup and discuss the results obtained with the various configurations in Section 9. Finally we sum up our findings in Section 10.

3 Related Work

Hierarchical phrase-based translation has been pioneered by David Chiang (Chiang, 2005) with his Hiero system. He induces a weighted synchronous context-free grammar from parallel text, the search is typically carried out using the cube pruning algorithm.

Soft syntactic labels. Soft syntactic labels have been first introduced by Venugopal et al. (2009) as an extension to their previous SAMT approach. In SAMT, the generic non-terminal of the hierarchical model is substituted with syntactic categories. Using soft syntactic labels, these additional non-terminals are considered in a probabilistic way, no hard constraints are imposed. Many other groups have presented similar approaches to augment hierarchical systems with syntactic information recently, e.g. Chiang (2010), Hoang and Koehn (2010), Stein et al. (2010), and Baker et al. (2010), among others. Results on Arabic-English tasks are rarely reported.

Discriminative word lexicon. Several variants of discriminatively trained extended lexicon models have been utilized effectively within quite different statistical machine translation systems. Mauser et al. (2009) integrate a discriminative as well as a trigger-based extended lexicon model into a phrase-based system, Huck et al. (2010) report results within hierarchical decoding, and Jeong et al. (2010) use a discriminative lexicon model with morphological and dependency features in a treelet translation system.

Reordering extensions. Some techniques to manipulate the reordering capabilities of hierarchical systems by modifying the grammar have been published lately. Iglesias et al. (2009) investigate a maximum phrase jump of 1 (MJ1) reordering model. They include a swap rule, but withdraw all hierarchical phrases. He et al. (2010) combine an additional BTG-style swap rule with a maximum entropy based lexicalized reordering model and achieve improvements on a Chinese-English task. Vilar et al. (2010) apply IBM-style reordering enhancements successfully to a German-English Europarl task.

Shallow rules. The way to restrict the parsing depth we apply in this work has been introduced by Iglesias et al. (2009), along with methods to filter the hierarchical rule set.

Lightly-supervised training. Large-scale lightly-supervised training for SMT as we define it in this paper has been introduced by Schwenk (2008). Schwenk automatically translates a large amount of monolingual data with an initial Moses (Koehn et al., 2007) baseline system from French into English. He uses the resulting unsupervised bitexts as additional training corpora to improve the baseline system. In Schwenk's original work, an additional bilingual dictionary is added to the baseline. With lightly-supervised training, Schwenk achieves improvements of around one BLEU point over the baseline. In a later work (Schwenk and Senellart, 2009) he applies the same method for translation model adaptation on an Arabic-French task. We extend this line of research by investigating the impact of lightly-supervised training across different SMT systems and translation paradigms.

4 Soft Syntactic Labels

A possibility to enhance the hierarchical model is to extend the set of non-terminals from the original generic symbol to a richer, syntax-oriented set. However, augmenting the set of non-terminals also restricts the parsing space and thus we alter the set of possible translations. Furthermore, it can happen that no parse can be found for some input sentences. To address this issue, our extraction is extended in a similar way as in the work of Venugopal et al. (2009): for every rule in the grammar, we store information about the possible non-terminals that can be substituted in place of the generic non-terminal X, together with a probability for each combination of non-terminal symbols (cf. Figure 1).

During decoding, we compute two additional quantities for each derivation d. The first one is denoted by $p_h(Y|d)$ (h for "head") and reflects the probability that the derivation d under consideration of the additional non-terminal symbols has Y as its starting symbol. This quantity is needed for computing the probability $p_{\text{syn}}(d)$ that the derivation conforms with the extended set of non-terminals. Let r be the top rule in derivation d, with n non-terminal symbols. For each of these non-terminal symbols we substitute the sub-derivations $d_1, \ldots, d_n$ in r. Denoting with S the extended set of non-terminals, $p_{\text{syn}}(d)$ is defined as

$$p_{\text{syn}}(d) = \sum_{s \in S^{n+1}} \left(p(s|r) \cdot \prod_{k=2}^{n+1} p_h(s[k]|d_{k-1}) \right) . \tag{1}$$

We use the notation $[\cdot]$ to address the elements of a vector.

The probability p_h is computed in a similar way, but the summation index is restricted only to those vectors of non-terminal substitutions where the left-hand side is the one for which we want to compute the probability:

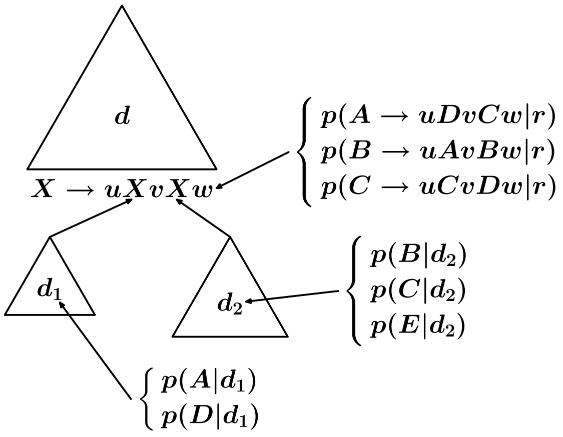

Figure 1: Visualization of the soft syntactic labels approach (Section 4). For each derivation, the probabilities of non-terminal labels are computed.

$$p_h(Y|d) = \sum_{s \in S^{n+1}: s[1]=Y} \left(p(s|r) \cdot \prod_{k=2}^{n+1} p_h(s[k]|d_{k-1}) \right) . \tag{2}$$

5 Discriminative Word Lexicon

We integrate a discriminative word lexicon (DWL) model that is very similar to the one presented by Mauser et al. (2009). This type of extended lexicon model accounts for global source sentence context to make predictions of target words. It goes beyond the capabilities of the standard model set of typical hierarchical systems as word lexicons and phrase models (even with hierarchical phrases) normally do not consider context beyond the phrase boundaries.

The DWL model acts as a classifier that predicts the words contained in the translation from the words given in the source sentence. The sequential order or any other structural interdependencies between the words on the source side as well as on the target side are ignored.

Let V_F be the source vocabulary and V_E be the target vocabulary. Then, we represent the source side as a bag of words by employing a count vector $\mathbf{F} = (\ldots, F_f, \ldots)$ of dimension $|V_F|$, and the target side as a set of words by employing a binary vector $\mathbf{E} = (\ldots, E_e, \ldots)$ of dimension $|V_E|$. Note that F_f is a count and E_e is a bit. The model estimates the probability $p(\mathbf{E}|\mathbf{F})$, i.e. that the target sentence consists of a set of target words given a bag of source words. For that purpose, individual models $p(E_e|\mathbf{F})$ are trained for each target word $e \in V_E$ (i.e. target word e should be included in the sentence, or not), which decomposes the problem into many separate two-class classification problems in the way shown in Equation (3).

$$p(\mathbf{E}|\mathbf{F}) = \prod_{e \in V_E} p(E_e|\mathbf{F}) \tag{3}$$

Each of the individual classifiers is modeled as a log-linear model

$$p(E_e|\mathbf{F}) = \frac{e^{g(E_e,\mathbf{F})}}{\sum\limits_{\tilde{E}_e \in \{0,1\}} e^{g(\tilde{E}_e,\mathbf{F})}} \tag{4}$$

with the function

$$g(E_e, \mathbf{F}) = E_e \lambda_e + \sum_{f \in V_F} E_e F_f \lambda_{ef}\,, \tag{5}$$

where the λ_{ef} represent lexical weights and the λ_e are prior weights. Though the log-linear model offers a high degree of flexibility concerning the kind of features that may be used, we simply use the source words as features. The feature weights for the individual classifiers are trained with the improved RProp+ algorithm (Igel and Hüsken, 2003).

6 IBM-style Reorderings for Hierarchical Phrase-based Translation

We extend the hierarchical phrase-based system with a jump model as proposed by Vilar et al. (2010), to permit jumps across whole blocks of symbols, and to facilitate a less restricted placement of phrases within the target sequence. The model is made up of additional, non-lexicalized rules and a distance-based jump cost, and allows for constrained reorderings. It is comparable to conventional phrase-based IBM-style reordering (Zens et al., 2004).

The hierarchical model comprises hierarchical rules with up to two non-neighboring non-terminals on their right-hand side as built-in reordering mechanism. An initial rule

$$S \rightarrow \langle X^{\sim 0}, X^{\sim 0} \rangle \tag{6}$$

is engrafted, as well as a special *glue rule* that the system can use for serial concatenation of phrases as in monotonic phrase-based translation (Chiang, 2005):

$$S \rightarrow \langle S^{\sim 0} X^{\sim 1}, S^{\sim 0} X^{\sim 1} \rangle \tag{7}$$

S denotes the start symbol of the grammar, the X symbol is a generic non-terminal which is used on all left-hand sides of the rules that are extracted from the training corpus and as a placeholder for the gaps within the right-hand side of hierarchical rules. $^\sim$ defines a one-to-one relation between the non-terminals within the source part and the non-terminals within the target part of hierarchical rules.

To enable IBM-style reorderings with a window length of 1, we replace the two rules from Equations (6) and (7) by the rules given in Equation (8):

$$\begin{aligned}
S &\rightarrow \langle M^{\sim 0}, M^{\sim 0} \rangle \\
S &\rightarrow \langle M^{\sim 0} S^{\sim 1}, M^{\sim 0} S^{\sim 1} \rangle & \dagger \\
S &\rightarrow \langle B^{\sim 0} M^{\sim 1}, M^{\sim 1} B^{\sim 0} \rangle & \ddagger \\
M &\rightarrow \langle X^{\sim 0}, X^{\sim 0} \rangle \\
M &\rightarrow \langle M^{\sim 0} X^{\sim 1}, M^{\sim 0} X^{\sim 1} \rangle & \dagger \\
B &\rightarrow \langle X^{\sim 0}, X^{\sim 0} \rangle \\
B &\rightarrow \langle B^{\sim 0} X^{\sim 1}, B^{\sim 0} X^{\sim 1} \rangle & \dagger
\end{aligned} \tag{8}$$

In these rules, the M non-terminal represents a block that will be translated in a monotonic way, and the B is a "back jump". Although these two symbols could be joined into one (the production rules are the same for both), it is useful to keep them separate to facilitate the computation of the distortion costs. The reordering extensions can easily be adapted to the shallow grammar that will be described in the following section.

We add a binary feature that fires for the rules that act analogous to the glue rule (†). Additionally, a distance penalty based on the jump width is computed during decoding when the back jump rule (‡) is applied.

7 Deep Rules vs. Shallow Rules

In order to constrain the search space of the decoder, we can modify the grammar so that the depth of the hierarchical recursion is restricted to one (Iglesias et al., 2009).

We replace the generic non-terminal X by two distinct non-terminals XH and XP. By changing the left-hand sides of the rules, we allow lexical phrases only to be derived from XP, and hierarchical phrases only from XH. On all right-hand sides of hierarchical rules, the X is replaced by XP. Gaps within hierarchical phrases can thus only be filled with purely lexicalized phrases, but not a second time with hierarchical phrases.

Note that the initial rule (Eqn. 6) has to be substituted with

$$\begin{aligned} S &\rightarrow \langle XP^{\sim 0}, XP^{\sim 0} \rangle \\ S &\rightarrow \langle XH^{\sim 0}, XH^{\sim 0} \rangle \, , \end{aligned} \tag{9}$$

and the glue rule (Eqn. 7) has to be substituted with

$$\begin{aligned} S &\rightarrow \langle S^{\sim 0} XP^{\sim 1}, S^{\sim 0} XP^{\sim 1} \rangle \\ S &\rightarrow \langle S^{\sim 0} XH^{\sim 1}, S^{\sim 0} XH^{\sim 1} \rangle \, . \end{aligned} \tag{10}$$

We refer to this kind of rule set and the parses produced with such a grammar as *shallow*, in contrast to the standard rule set and parses which we denote as *deep*.

8 Improving the Translation Model with Lightly-supervised Training

In this section, we propose a novel cross-system and cross-paradigm variant of lightly-supervised training. More specifically, we extend the translation model of the hierarchical system using unsupervised parallel training data derived from automatic translations produced with a conventional phrase-based system. The additional bitexts are created by translating large amounts of monolingual source language data with a conventional phrase-based system. Word alignments are trained to be able to extract phrases from the data. Note that, unlike Schwenk (2008), we do not try to improve the same system which was used to create the unsupervised data but rather change the translation paradigm, in order to combine the strengths of both approaches.

Conventional phrase-based systems are usually able to correctly translate short sequences in a local context, but often have problems in producing a fluent sentence structure across long distances Thus, we decided to include lexical phrases from the unsupervised data, but to restrict the set of phrases with non-terminals to those that were derived from the more reliable human-generated parallel data.

To our knowledge, this is the first time that lightly-supervised training is applied to a hierarchical system.

9 Experiments

We use the open source Jane toolkit (Vilar et al., 2010) for our experiments, a hierarchical phrase-based translation software written in C++. We give a detailed description of our setup to ease reproduction by the scientific community.

9.1 Experimental Setup

The phrase table of the baseline system has been produced from a parallel training corpus of 2.5M Arabic-English sentence pairs. Word alignments in both directions were trained with GIZA++ and symmetrized according to the refined method that was proposed by Och and Ney (2003). To reduce the size of the phrase table, a minimum count cutoff of one and an extraction pruning threshold of 0.1 have been applied to hierarchical phrases.

	Arabic	English
Sentences	2 514 413	
Running words	54 324 372	55 348 390
Vocabulary	264 528	207 780
Singletons	115 171	91 390

Table 1: Data statistics for the preprocessed Arabic-English parallel training corpus. In the corpus, numerical quantities have been replaced by a special category symbol.

The models integrated into our baseline system are: phrase translation probabilities and lexical translation probabilities at phrase level, each for both translation directions, length penalties on word and phrase level, three binary features for hierarchical phrases, glue rule, and rules with non-terminals at the boundaries, a binary feature that fires if the phrase has a source length of only one word, three binary features marking phrases that have been seen at least two, four, or six times, respectively, and an n-gram language model.

Our setups use a 4-gram language model with modified Kneser-Ney smoothing. It was created with the SRILM toolkit (Stolcke, 2002) and was trained on a large collection of monolingual data including the target side of the parallel corpus and the LDC Gigaword v4 corpus. We measured a perplexity of 96.9 on the four reference translations of MT06.

The scaling factors of the log-linear model combinations have been optimized with MERT on the MT06 NIST test corpus. MT08 was employed as held-out test data. Detailed statistics about the parallel training data are given in Table 1, for the development and the test corpus in Table 2.

To obtain the syntactic annotation for the soft syntactic labels, the Berkeley Parser (Petrov et al., 2006) has been applied.

The DWL model has been trained on a manually selected high-quality subset of the parallel data of

	dev (MT06)	test (MT08)
Sentences	1 797	1 360
Running words	49 677	45 095
Vocabulary	9 274	9 387
OOV [%]	0.5	0.4

Table 2: Data statistics for the preprocessed Arabic part of the dev and test corpora. In the corpus, numerical quantities have been replaced by a special category symbol.

277 234 sentence pairs. The number of features per target word which are considered during training is equal to the size of the source vocabulary of the training corpus, i.e. 122 592 in this case. We carried out 100 training iterations per target word with the improved RProp+ algorithm. After training, the full DWL model was pruned with a threshold of 0.1. The pruned model contains on average 80 features per target word.

9.2 Unsupervised Data

The unsupervised data that we integrate has been created by automatic translations of parts of the Arabic LDC Gigaword corpus (mostly from the HYT collection) with a conventional phrase-based system. Translating the monolingual Arabic data has been performed by LIUM, Le Mans, France. We thank Holger Schwenk for kindly providing the translations.

The score computed by the decoder for each translation has been normalized with respect to the sentence length and used to select the most reliable sentence pairs. We report the statistics of the unsupervised data in Table 3. Word alignments for the unsupervised data have been produced in the same way as for the baseline bilingual training data.

	Arabic	English
Sentences	4 743 763	
Running words	121 478 207	134 227 697
Vocabulary	306 152	237 645
Singletons	130 981	102 251

Table 3: Data statistics for the Arabic-English unsupervised training corpus after selection of the most reliable sentence pairs. In the corpus, numerical quantities have been replaced by a special category symbol.

Using the unsupervised data in the way described in Section 8 increases the number of non-hierarchical phrases by roughly 30%, compared to the baseline system where the phrase table is extracted from the human-generated bitexts only.

9.3 Translation Results

The empirical evaluation of all our systems is presented in Table 4. All methods are evaluated on the two standard metrics BLEU and TER and checked for statistical significance over the baseline. The confidence intervals have been computed using bootstrapping for BLEU and Cochran's approximate ratio variance for TER (Leusch and Ney, 2009). We report experimental results on both the development and the test corpus (MT06 and MT08, respectively). The figures with deep and with shallow rules are set side by side in separate columns to facilitate a direct comparison between them. All the setups given in separate rows exist in a deep and a shallow variant.

One of the objectives is to compare the deep and shallow setups. This has an important effect in practice, as the shallow setup is much more efficient in terms of computational effort, with speedups of 5 to 10 when compared to the (standard) deep setup. We found that the shallow system translation quality is comparable to the deep system.

The inclusion of the unsupervised data leads to a gain on the unseen test set of +0.7% BLEU / -0.6% TER absolute in the deep setup and +0.8% BLEU / -0.2% TER absolute in the shallow setup. This shows that the proposed approach is beneficial and allows to use available monolingual data to improve the performance of the system.

A further clear increase in translation quality is achieved by adding the extended word lexicon model. Both the deep and the shallow setup benefit from the incorporation of the discriminative word lexicon, with gains of about the same order of magnitude (+0.7% BLEU / -0.7% TER with deep rules, +0.6% BLEU / -1.0% TER with shallow rules). Combining the unsupervised training data and the extended word lexicon we arrive at an improvement that is significant at the 95% confidence level.

The two other approaches investigated in this paper do not really help improving the translation quality. The syntactic labels improve the BLEU score only slightly in the deep approach, and even degrade the translation quality in the shallow setup. The additional reorderings have nearly

	dev (MT06)				test (MT08)			
	deep		shallow		deep		shallow	
	BLEU [%]	TER [%]	BLEU [%]	TER [%]	BLEU [%]	TER [%]	BLEU [%]	TER [%]
HPBT Baseline	43.9	50.2	44.1	49.9	44.3$_{\pm 1.1}$	50.0$_{\pm 0.9}$	44.4$_{\pm 1.1}$	49.4$_{\pm 0.9}$
+ Unsup	45.2	48.9	45.1	49.1	45.0	49.4	45.2	49.2
+ Unsup + DWL	45.8	48.3	45.8	48.4	**45.7**	**48.7**	**45.8**	**48.2**
+ Unsup + Syntactic Labels	45.1	49.0	45.2	49.1	45.2	49.3	45.0	49.0
+ Unsup + Reorderings	45.4	48.8	45.3	49.0	45.3	49.1	45.3	48.9
+ Unsup + DWL + Syntactic Labels	46.2	48.0	46.1	48.2	**46.0**	**48.2**	**45.8**	**48.3**
+ Unsup + DWL + Reorderings	46.1	47.9	46.1	48.2	**45.7**	**48.7**	**45.9**	**48.2**

Table 4: Results for the NIST Arabic-English translation task (truecase). The 95% confidence interval is given for the baseline systems. Results in bold are significantly better than the baseline.

no effect on the translation.

These results, although a bit disappointing, were to be expected. As stated above, the Arabic-English language pair is rather monotonic and these two last approaches are more useful when dealing with translation directions where the word order in the languages is rather different. The degradation in translation quality in the shallow setup can be explained by the restriction in the parse trees that are constructed during the translation process. By restricting their depth they can not conform with the syntax trees derived from linguistic parsing.

The best results are obtained with a deep system including all the advanced methods at once, with the exception of the additional reorderings. It achieves an improvement of +1.7% BLEU / -1.8% TER over the baseline. For the shallow system, the combination of the methods does not improve over the unsupervised data and discriminative word lexicon alone. The final result does not exceed the translation quality of the best deep setup, but remember that the computation time is significantly decreased.

10 Conclusion

We presented a cross-system and cross-paradigm lightly-supervised training approach. We demonstrated that improving the non-hierarchical part of the translation model with lightly-supervised training is a very effective technique. On the NIST Arabic-English task, we evaluated various recently developed methods separately as well as in combination. Our results suggest that soft syntactic labels and IBM-style reordering extensions are less helpful. By including the discriminative word lexicon model, we have been able to increase the performance of the hierarchical system significantly. Our experiments with shallow rules confirm that a deep recursion for hierarchical rules is not essential to achieve competitive performance for the Arabic-English language pair, while dramatically decreasing the computational effort.

Acknowledgments

The authors would like to thank Holger Schwenk from LIUM, Le Mans, France, for making the automatic translations of the Arabic LDC Gigaword corpus available. This work was partly realized as part of the Quaero Programme, funded by OSEO, French State agency for innovation, and also partly based upon work supported by the Defense Advanced Research Projects Agency (DARPA) under Contract No. HR0011-08-C-0110. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the DARPA.

References

Baker, Kathryn, Michael Bloodgood, Chris Callison-Burch, Bonnie Dorr, Nathaniel Filardo, Lori Levin, Scott Miller, and Christine Piatko. 2010. Semantically-Informed Syntactic Machine Translation: A Tree-Grafting Approach. In *Conf. of the Assoc. for Machine Translation in the Americas (AMTA)*, Denver, CO, October/November.

Birch, Alexandra, Phil Blunsom, and Miles Osborne. 2009. A Quantitative Analysis of Reordering Phenomena. In *Proc. of the Workshop on Statistical Machine Translation*, pages 197–205, Athens, Greece, March.

Chiang, David. 2005. A Hierarchical Phrase-Based Model for Statistical Machine Translation. In *Proc. of the 43rd Annual Meeting of the Assoc. for Computational Linguistics (ACL)*, pages 263–270, Ann Arbor, MI, June.

Chiang, David. 2010. Learning to Translate with Source and Target Syntax. In *Proc. of the Annual Meeting of the Assoc. for Computational Linguistics (ACL)*, pages 1443–1452, Uppsala, Sweden, July.

de Gispert, Adrià, Gonzalo Iglesias, Graeme Blackwood, Eduardo R. Banga, and William Byrne. 2010. Hierarchical Phrase-Based Translation with Weighted Finite-State Transducers and Shallow-n Grammars. *Computational Linguistics*, 36(3):505–533.

He, Zhongjun, Yao Meng, and Hao Yu. 2010. Extending the Hierarchical Phrase Based Model with Maximum Entropy Based BTG. In *Conf. of the Assoc. for Machine Translation in the Americas (AMTA)*, Denver, CO, October/November.

Hoang, Hieu and Philipp Koehn. 2010. Improved Translation with Source Syntax Labels. In *ACL 2010 Joint Fifth Workshop on Statistical Machine Translation and Metrics MATR*, pages 409–417, Uppsala, Sweden, July.

Huck, Matthias, Martin Ratajczak, Patrick Lehnen, and Hermann Ney. 2010. A Comparison of Various Types of Extended Lexicon Models for Statistical Machine Translation. In *Conf. of the Assoc. for Machine Translation in the Americas (AMTA)*, Denver, CO, October/November.

Igel, Christian and Michael Hüsken. 2003. Empirical Evaluation of the Improved Rprop Learning Algorithm. *Neurocomputing*, 50:2003.

Iglesias, Gonzalo, Adrià de Gispert, Eduardo R. Banga, and William Byrne. 2009. Rule Filtering by Pattern for Efficient Hierarchical Translation. In *Proc. of the 12th Conf. of the Europ. Chapter of the Assoc. for Computational Linguistics (EACL)*, pages 380–388, Athens, Greece, March.

Jeong, Minwoo, Kristina Toutanova, Hisami Suzuki, and Chris Quirk. 2010. A Discriminative Lexicon Model for Complex Morphology. In *Conf. of the Assoc. for Machine Translation in the Americas (AMTA)*, Denver, CO, October/November.

Koehn, P., H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, et al. 2007. Moses: Open Source Toolkit for Statistical Machine Translation. In *Proc. of the Annual Meeting of the Assoc. for Computational Linguistics (ACL)*, pages 177–180, Prague, Czech Republic, June.

Leusch, Gregor and Hermann Ney. 2009. Edit distances with block movements and error rate confidence estimates. *Machine Translation*, December.

Mauser, Arne, Saša Hasan, and Hermann Ney. 2009. Extending Statistical Machine Translation with Discriminative and Trigger-Based Lexicon Models. In *Proc. of the Conf. on Empirical Methods for Natural Language Processing (EMNLP)*, pages 210–218, Singapore, August.

Och, Franz Josef and Hermann Ney. 2003. A Systematic Comparison of Various Statistical Alignment Models. *Computational Linguistics*, 29(1):19–51, March.

Petrov, Slav, Leon Barrett, Romain Thibaux, and Dan Klein. 2006. Learning Accurate, Compact, and Interpretable Tree Annotation. In *Proc. of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Assoc. for Computational Linguistics*, pages 433–440, Sydney, Australia, July.

Schwenk, Holger and Jean Senellart. 2009. Translation Model Adaptation for an Arabic/French News Translation System by Lightly-Supervised Training. In *MT Summit XII*, Ottawa, Ontario, Canada, August.

Schwenk, Holger. 2008. Investigations on Large-Scale Lightly-Supervised Training for Statistical Machine Translation. In *Proc. of the Int. Workshop on Spoken Language Translation (IWSLT)*, pages 182–189, Waikiki, Hawaii, October.

Stein, Daniel, Stephan Peitz, David Vilar, and Hermann Ney. 2010. A Cocktail of Deep Syntactic Features for Hierarchical Machine Translation. In *Conf. of the Assoc. for Machine Translation in the Americas (AMTA)*, Denver, CO, October/November.

Stolcke, Andreas. 2002. SRILM – an Extensible Language Modeling Toolkit. In *Proc. of the Int. Conf. on Spoken Language Processing (ICSLP)*, volume 3, Denver, CO, September.

Venugopal, Ashish, Andreas Zollmann, N.A. Smith, and Stephan Vogel. 2009. Preference Grammars: Softening Syntactic Constraints to Improve Statistical Machine Translation. In *Proc. of the Human Language Technology Conf. / North American Chapter of the Assoc. for Computational Linguistics (HLT-NAACL)*, pages 236–244, Boulder, CO, June.

Vilar, David, Daniel Stein, Matthias Huck, and Hermann Ney. 2010. Jane: Open Source Hierarchical Translation, Extended with Reordering and Lexicon Models. In *ACL 2010 Joint Fifth Workshop on Statistical Machine Translation and Metrics MATR*, pages 262–270, Uppsala, Sweden, July.

Zens, Richard, Hermann Ney, Taro Watanabe, and Eiichiro Sumita. 2004. Reordering Constraints for Phrase-Based Statistical Machine Translation. In *COLING '04: The 20th Int. Conf. on Computational Linguistics*, pages 205–211, Geneva, Switzerland, August.