A Comparison of Various Types of Extended Lexicon Models for Statistical Machine Translation

Matthias Huck, Martin Ratajczak, Patrick Lehnen and Hermann Ney
Lehrstuhl für Informatik 6
RWTH Aachen University
Aachen, Germany
{huck,ratajczak,lehnen,ney}@cs.rwth-aachen.de

Abstract

In this work we give a detailed comparison of the impact of the integration of discriminative and trigger-based lexicon models in state-of-the-art hierarchical and conventional phrase-based statistical machine translation systems. As both types of extended lexicon models can grow very large, we apply certain restrictions to discard some of the less useful information. We show how these restrictions facilitate the training of the extended lexicon models. We finally evaluate systems that incorporate both types of models with different restrictions on a large-scale translation task for the Arabic-English language pair. Our results suggest that extended lexicon models can be substantially reduced in size while still giving clear improvements in translation performance.

1 Introduction

Taking long-range dependencies into account is still one of the main problems in today's statistical machine translation (SMT). State-of-the-art systems comprise components like a phrase translation model and n-gram language models that act effectively within a local context and give reliable results as long as only information from a limited window is required. But reordering in translation between different languages, recursive embedding of subphrases, as it is common in natural language, and distant lexical interconnections are hard to model and difficult to handle in a computationally efficient way. [1]

The hierarchical phrase-based approach to SMT promises to be able to capture translations whose scope is larger than a few consecutive words (Chiang, 2005; Chiang, 2007). By allowing gaps within bilingual phrases that are indicated by corresponding place-holders (i.e. co-indexed non-terminals), the phrase table of a hierarchical phrase-based translation (HPBT) system can be considered to be the production set of a synchronous context-free grammar. This formal grammar usually does not comply with a linguistically motivated grammar, but as the search procedure is realized as a probabilistic parser, the hierarchical phrase-based paradigm connects somewhat closer to more linguistics-related work in natural language processing than conventional phrase-based translation (PBT). Several efforts have been made recently to engineer syntactically more informed SMT systems. Appropriate models can be introduced into the log-linear framework of modern SMT systems (Och and Ney, 2002) without having to impose any hard contraints on the translation process. Bringing different lines of research together in a natural way by augmenting hierarchical translation with syntactic knowledge has primarily been done with the intent to be able to produce better structured outputs with the resulting systems.

On the other hand, conventional phrase-based translation with left-to-right target generation has proven very successful and robust by relying on statistics learned purely on surface forms from huge corpora. Such systems still outperform hierarchical setups in many evaluations. (Galley and Manning, 2010) even show that conventional systems can be

[1] (Knight, 1999) proofs that the decoding problem with unrestricted reorderings is NP-complete.

extended in a way that they are able to make use of phrases with gaps similar to the rule set of hierarchical systems. In their experiments, a conventional system with gappy phrases and lexicalized reordering produces a significantly better output for Chinese-English than a hierarchical one without any syntactic enhancements.

(Auli et al., 2009) challenge the common assumption that there are structural differences in the types of outputs the two translation approaches can produce. Analyzing the search spaces of conventional phrase-based and hierarchical systems, they find a high overlap. They argue that the main difference is in the parameterization, not in the expressiveness of the translation models.

Recent research has demonstrated how two types of extended lexicon models called *triplet lexicon model* (we will abbreviate this simply as *triplets* in many cases) and *discriminative word lexicon* (DWL) can improve the translation results of conventional phrase-based systems in n-best reranking as well as directly in beam-search decoding (Mauser et al., 2009). Both of them account for global source sentence context to predict context-specific target words. Their main advantage is that they promote a better lexical selection than the baseline models alone are able to achieve.

With the availability of DWL and triplet model scoring implementations in a state-of-the-art hierarchical phrase-based translation system (Vilar et al., 2010), we are now in a position to compare conventional and hierarchical phrase-based setups — either of them enriched with extended lexicon models — against each other.

On the large-scale NIST Arabic-English translation task, we show that though a gap between the BLEU scores of the baseline systems can be observed, the two paradigms perform exactly the same if triplet and DWL models are added to the setups. Hierarchical and standard phrase-based statistical machine translation currently seem to operate at a comparable level, with advantages in some points for each of them. A good parameterization is a crucial aspect by all means.

Regrettably it is barely feasible to make use of the full bilingual data that is available for language pairs of wide interest like Arabic-English for the training of triplet and discriminative word lexicon models. The high computational demands compel to work on corpora of a smaller size. Moreover, the trained models are usually large, therefore yielding a noticeable increase of memory requirements and runtime during decoding. We investigate methods to tackle these problems and examine in which ways extended lexicon models can be restricted while still retaining the most useful information they provide.

2 Overview

In Section 3, we give a short overview of the previously published work this paper builds on. We introduce triplet lexicon and discriminative word lexicon models in Section 4 and describe the modifications we apply to reduce their computational demands in training, and their final size.

The experimental evaluation is presented in Section 5. We first give a characterization of the experimental setup and the main details of our systems. We then report on the different extended lexicon models we trained and proceed with a comparison of the translation results using these models in standard phrase-based and hierarchical translation.

3 Previous Work

(Hasan et al., 2008) proposed triplet lexicon models for statistical machine translation for the first time. Triplet lexicon models are related to the well-known IBM-1 model (Brown et al., 1993) but extend it with a second trigger. (Hasan et al., 2008) also introduced the restrictions that are applied to triplets in this work, they did however apply the models only in an n-best list reranking framework. They evaluated their methods on a small Chinese-English and on a Spanish-English/English-Spanish task.

(Hasan and Ney, 2009) investigated triplet lexicon scoring in a conventional phrase-based decoder and compared translation performance of the so-called path-constrained (or path-aligned) triplet models applied in reranking to an integrated application in search on a large-scale Chinese-English task. They did not evaluate different variants of the model.

The DWL model in a variant that is trained using seen features as well as unseen features was presented by (Mauser et al., 2009). We will compare our new variant of DWL models to the model as described by them. (Mauser et al., 2009) also compared the effect of a triplet and a DWL model in phrase-based decoding on a Chinese-English task and on the Arabic-English task that we likewise work on. They did not evaluate different variants of the two extended lexicon models, nor did they apply them in a hierarchical phrase-based system.

Implementations of triplet and DWL scoring functionality in a hierarchical decoder were published by (Vilar et al., 2010) recently.

4 Extended Lexicon Models with and without Restrictions

4.1 Triplet Lexicon

The triplet lexicon relies on triplets which are composed of two source language words triggering one target language word, i.e. it models probabilities $p(e|f,f')$. The probability of a whole target sentence e_1^I given the source sentence f_1^J is thus calculated as

$$\begin{aligned} p(e_1^I|f_1^J) &= \prod_{i=1}^{I} p(e_i|f_1^J) \\ &= \prod_{i=1}^{I} \frac{2}{J(J+1)} \sum_{j=0}^{J} \sum_{j'=j+1}^{J} p(e_i|f_j,f_{j'}). \end{aligned} \quad (1)$$

The so-called *path-constrained* (or *path-aligned*) triplet model variant restricts the first trigger f to the aligned target word e. The second trigger f' is allowed to range over all remaining words of the source sentence. When $\{a_{ij}\}$ denotes the alignment matrix of the sentence pair e_1^I and f_1^J, the probability of a whole target sentence results in

$$\begin{aligned} p(e_1^I|f_1^J,\{a_{ij}\}) &= \prod_{i=1}^{I} p(e_i|f_1^J,\{a_{ij}\}) \\ &= \prod_{i=1}^{I} \frac{1}{Z_i} \sum_{j\in\{a_i\}} \sum_{j'=1}^{J} p(e_i|f_j,f_{j'}). \end{aligned} \quad (2)$$

The double summation is normalized with the factor $Z_i = J \cdot |\{a_i\}|$. $j \in \{a_i\}$ expresses that f_j is aligned to the current target word e_i.

To further reduce the size of a triplet model, count cutoffs can be applied. This means that triplets that occur less than a fixed number of times in the corpus are not considered in the training of the model.

Like the IBM-1 model, triplets are trained iteratively with the Expectation-Maximization (EM) algorithm (Dempster et al., 1977).

4.2 Discriminative Word Lexicon

The discriminative word lexicon (DWL) model estimates the probability that the target sentence consists of a set of target words $\mathbf{e}$ given a set of words $\mathbf{f}$ in the source sentence. The set of target words $\mathbf{e}$ can be coded in a binary vector $\mathbf{E} = (..., E_e, ...)$. The indicator variable E_e is set to one if the word e is contained in the target sentence, otherwise it is set to zero. In the same way the counts F_f of the source words can be represented as a count vector $\mathbf{F} = (..., F_f, ...)$. Interdependencies between the words on the target side as well as on the source side are ignored. Thus the probability for the whole sentence is made up of the individual and independent probabilities over the target vocabulary V_E

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

The probability for a single target word is modeled as a log-linear model

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

with the function

$$g(E_e,\mathbf{F}) = E_e\lambda_e + \sum_f E_e F_f \lambda_{ef} \quad (5)$$

where λ_{ef} represent the lexical weights and λ_e are the prior weights.

Due to the independence of the probabilities $p(E_e|\mathbf{F})$ for each target word, it is easy to parallelize the training procedure. The models are trained with the improved RProp+ (Igel and Hüsken, 2003) in contrast to (Mauser et al., 2009) where the L-BFGS method is used. For each target word 100 iterations of the training algorithm are carried out. Regularization is done using Gaussian priors.

4.2.1 Feature Selection with Binary Training Criterion

One straightforward way to reduce the size of a DWL model is to apply threshold pruning to the features associated with each target word. However, this does not cut down the computational resources needed for training as the lexical weights have to be obtained first.

To reduce the training time we train only seen pairs (e, f) and set the parameters λ_{ef} for the unseen pairs to zero. This approach is called sparse DWL model in contrast to the full DWL model where we train parameters λ_{ef} for both the seen and unseen pairs. Let S be the set of seen pairs, then the function from Equation (5) changes to

$$g(E_e, \mathbf{F}) = E_e\lambda_e + \sum_{f:(e,f)\in S} E_e F_f \lambda_{ef}. \quad (6)$$

Though training only seen pairs as in the sparse variant of the model already greatly reduces the effort, supplementary techniques for an early restriction of the number of features would be beneficial.

We therefore present an approach for feature selection through estimation of the gain in conditional log-likelihood

$$\Delta G_{e\hat{f}} = \sum_n \log \frac{p_{\hat{f}}(E_e|\mathbf{F})}{p_0(E_e|\mathbf{F})} \quad (7)$$

by adding a new feature $(e, \hat{f})$ to a basic log-linear model $p_0(E_e|\mathbf{F})$. Our approach has been motivated by the work on feature induction for random fields by (Della Pietra et al., 1997). In contrast to that work a binary training criterion has been used to match the criterion for the DWL model. We use the gain as ranking criterion. For every target word e we get a ranked list of features and select the n best of them for the training of the sparse DWL model.

Let us assume for now that we have a basic log-linear model $p_0(E_e|\mathbf{F})$. Adding a new feature $(e, \hat{f})$ will result in an additional term in nominator and denominator of the new probability

$$p_{\hat{f}}(E_e|\mathbf{F}) = \frac{e^{E_e F_{\hat{f}} \lambda_{e\hat{f}}} p_0(E_e|\mathbf{F})}{\sum\limits_{\tilde{E}_e \in \{0,1\}} e^{\tilde{E}_e F_{\hat{f}} \lambda_{e\hat{f}}} p_0(\tilde{E}_e|\mathbf{F})}. \quad (8)$$

By means of the approximation $\log(1 + x) \approx x$ for small x and the restriction to binary features F_f the information gain can be approximated. When the base model $p_0(E_e|\mathbf{F})$ contains only one feature it is possible to calculate its parameter analytically. In our case the basic log-linear model $p_0(E_e|\mathbf{F})$ has only one parameter for the prior λ_e and can be calculated by using the logarithm of the relative frequencies. Then we maximize the approximated gain with respect to the parameter $\lambda_{e\hat{f}}$ to infer an approximated, but closed-form solution for its value. As result in this special case the information gain

$$\begin{aligned} \Delta G_{e\hat{f}} \quad &\geq \quad N_{e\hat{f}} \Big(\log \frac{N_{e\hat{f}}}{N_{\hat{f}}} \frac{1}{\frac{N_e}{N}} - 1 \Big) \quad (9) \\ &+ \quad N \log(1 + \frac{N_e}{N}). \end{aligned}$$

can be calculated simply by using the counts $N_{e\hat{f}}$, N_e and N_f computed from the corpus.

It should be mentioned that this criterion can be applied only to seen pairs. In the case of unseen pairs the logarithm is undefined.

5 Experiments

5.1 Experimental Setup

We used a training corpus of 2.5M Arabic-English sentence pairs to set up the hierarchical as well as the conventional phrase-based systems. Word alignments in both directions were produced with GIZA++ and symmetrized according to the refined method that was proposed by (Och and Ney, 2003).

	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. Numbers have been replaced by a special category symbol.

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

All of the configurations use the same 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

	dev (MT06)	test (MT08)
Sentences	1 797	1 360
Running words	49 677	45 095
Vocabulary	9 274	9 387
OOV [%]	0.46	0.35

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

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.

5.1.1 Hierarchical Systems

The hierarchical translation system we utilize has been developed at RWTH and has recently been released as open source software (Vilar et al., 2010). It implements the hierarchical phrase-based paradigm that has been introduced by (Chiang, 2005).

We performed shallow search as defined in (Iglesias et al., 2009), i.e. we did not allow substitutions of non-terminals by strings containing non-terminals again, and ran the cube pruning algorithm (Huang and Chiang, 2007) with 500-best generation. Furthermore, we configured observation histogram pruning at a value of 50.

Apart from the hierarchical phrase translation model, the language model and the extended lexicon models, the log-linear model combination of our systems comprises source-to-target and target-to-source phrase translation probabilities, IBM-1 source-to-target and target-to-source lexical translation probabilities, two features that account for some control about the application of hierarchical rules as opposed to initial rules, length penalties on word and phrase level and four binary features, essentially simple count features.

5.1.2 Phrase-Based Systems

Our standard phrase-based machine translation system operates in the way described by (Zens and Ney, 2008). Phrase translation and word lexicon models in both directions, phrase and word penalties, a binary model that indicates a source phrase length of 1, a distortion model and the language model are incorporated in the log-linear model combination. We use phrase level IBM reordering constraints (Zens et al., 2004).

5.1.3 Extended Lexicon Models

We trained triplet models and sparse DWL models on a manually selected high-quality subset of the parallel data of 717 133 sentences. A full DWL model was trained on an even smaller part of just 277 234 sentence pairs.

Triplet models. We prepared several triplet models of the variant denoted as path-constrained in Section 4.1 as well as of the variant denoted as unconstrained. The number of EM iterations has been 6 in all cases.

Four different path-constrained triplet models are considered, one without any count cutoff and three with cutoffs of 2, 3, and 4, respectively. Like for the word alignments used for the phrase extraction, we used symmetrized GIZA++ alignments in the training of the path-constrained triplet models.

We do not report on translation results for an unconstrained triplet model without count cutoff because the computational costs for the application of a large model like that in search would have been very high. Instead we trained unconstrained triplet models with cutoffs of 7 and 10, the former still being the triplet model with the maximum number of triplets.

Details on the sizes of the models and on the computional requirements for their training are shown in Table 3.

DWL models. We prepared a sparse DWL model without any feature selection and two sparse DWL models using the feature selection as presented in Section 4.2.1. The maximal number of features per target word has been set to 1 000 and 100, respectively. For comparison we also trained a full DWL model of the type as present in (Mauser et al., 2009). This model is denoted simply as *DWL* in the tables throughout this paper. Because training runtimes are considerably higher than for the sparse models, we used a smaller training corpus, as already mentioned above. After training, the full DWL model was pruned with a threshold of 0.1.

	no. of triplets	training time [h:min]	training mem. [GB]
Triplets (cutoff 7)	140 401 010	34:48	7.1
Triplets (cutoff 10)	98 792 441	32:53	4.8
path-constrained Triplets	128 640 058	3:11	11.0
path-constrained Triplets (cutoff 2)	44 953 477	2:27	3.8
path-constrained Triplets (cutoff 3)	27 109 368	2:29	2.2
path-constrained Triplets (cutoff 4)	20 222 988	2:27	1.6

Table 3: Sizes and computational demands in training for the triplet models.

	avg. no. of features per target word	avg. training time [s] per target word
DWL (full, pruned after training with threshold 0.1)	80 (unpruned: 122 592)	225
sparse DWL	510	64
sparse DWL (max. 1 000 features)	190	36
sparse DWL (max. 100 features)	61	32

Table 4: Average number of features per target word and average training time per target word for the DWL models. Note that the model denoted as *DWL* has been pruned after training with a threshold of 0.1. The number of features per target word which have to be considered during the training of this model is equal to the size of the source vocabulary of the training corpus, i.e. 122 592 in this case. The measured differences in runtime should be considered as rough approximations to the actual differences in computational demands as training has been carried out in a distributed environment where some hardware specifications and the load on the machines vary. They still give a clue about the required effort.

Details on the average number of features of each model and on the computional requirements for their training are given in Table 4.

5.2 Translation Results

The translation results of all our systems on the unseen test set and also on the development set are listed in Table 5.

We observe that the HPBT baseline system is 0.6 BLEU points worse on the test set than the PBT baseline.

The best result using a DWL model is in fact achieved with the model denoted as *DWL* which has been trained involving unseen features and pruned after training with a threshold of 0.1. The sparse DWL model and the sparse DWL model with selection of maximal 1 000 features give improvements close to that. The sparse DWL model with a maximum of 100 features is barely helpful to the PBT system but still gives some boost to the HPBT system. The HPBT system altogether profits a bit more of the additional models, but the relative differences between systems with different DWL models are rather consistent across the two SMT paradigms.

Looking at triplet models, we can observe that unconstrained triplets perform better in the HPBT system while path-constrained triplets are more helpful in conventional PBT setups. Count cutoffs of 3 or more do not make sense for path-constrained triplets at the amount of data we employed.

The best performing systems overall integrate triplet and DWL models at once. PBT and HPBT are exactly on par with a best result of 46.0% BLEU on MT08 each. If the models are combined, the path-contrained triplet variant seems to interact better with the DWL model.

	dev (MT06)				test (MT08)			
	HPBT		PBT		HPBT		PBT	
	BLEU	TER	BLEU	TER	BLEU	TER	BLEU	TER
Baseline	43.2	50.8	44.1	49.4	**44.1**	50.1	**44.7**	49.1
DWL	45.3	48.7	45.1	48.4	**45.6**	48.4	**45.6**	48.4
sparse DWL	44.9	49.3	44.8	48.8	45.4	48.8	45.3	48.7
sparse DWL (max. 1 000 features)	44.4	49.9	44.8	49.0	45.3	49.1	45.2	48.8
sparse DWL (max. 100 features)	44.5	49.4	44.6	49.0	45.1	49.1	44.8	49.0
Triplets (cutoff 7)	44.5	49.2	44.8	48.8	**45.6**	48.6	45.2	48.6
Triplets (cutoff 10)	44.4	49.1	44.6	49.2	45.3	48.8	44.9	49.0
path-constrained Triplets	44.3	49.4	44.7	49.1	44.9	49.3	45.3	48.7
path-constrained Triplets (cutoff 2)	44.2	49.6	44.8	48.9	44.8	49.3	**45.4**	48.8
path-constrained Triplets (cutoff 3)	43.4	50.0	44.5	49.3	44.1	49.8	45.0	49.1
path-constrained Triplets (cutoff 4)	43.5	50.6	44.5	49.5	43.8	50.2	44.9	49.3
DWL + Triplets (cutoff 10)	45.0	48.9	45.1	48.5	45.3	48.6	45.5	48.5
DWL + path-constrained Triplets	45.2	48.8	45.1	48.6	**46.0**	48.5	45.8	48.3
DWL + path-constrained Triplets (cutoff 2)	45.1	48.9	45.4	48.4	45.5	48.5	**46.0**	48.3

Table 5: Results for the NIST Arabic-English translation task. BLEU and TER results are in percentage.

6 Conclusions

We showed that the two types of extended lexicon models — the triplet lexicon as well as the discriminative word lexicon — yield nice improvements in both a conventional phrase-based statistical machine translation system and in a hierarchical phrase-based system. A gap between the BLEU scores of the baseline systems is diminished when well-trained extended lexicon models account for an appropriate parameterization.

In addition, we demonstrated that variations of the triplet lexicon and DWL models require much less computational effort but still significantly enhance translation performance.

Acknowledgments

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

Michael Auli, Adam Lopez, Hieu Hoang, and Philipp Koehn. 2009. A Systematic Analysis of Translation Model Search Spaces. In *Proceedings of the Fourth Workshop on Statistical Machine Translation*, pages 224–232, Athens, Greece, March. Association for Computational Linguistics.

Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. 1993. The Mathematics of Statistical Machine Translation: Parameter Estimation. *Computational Linguistics*, 19(2):263–311, June.

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

David Chiang. 2007. Hierarchical Phrase-Based Translation. *Computational Linguistics*, 33(2):201–228, June.

Stephen Della Pietra, Vincent Della Pietra, and John Lafferty. 1997. Inducing Features of Random Fields. *IEEE Trans. Pattern Anal. Mach. Intell.*, 19(4):380–393.

Arthur P. Dempster, Nan M. Laird, and Donald B. Rubin. 1977. Maximum Likelihood from Incomplete Data via the EM Algorithm. *J. Royal Statist. Soc. Ser. B*, 39(1):1–22.

Michel Galley and Christopher D. Manning. 2010. Accurate Non-Hierarchical Phrase-Based Translation. In *Proc. of the Human Language Technology Conf. / North American Chapter of the Association for Computational Linguistics (HLT-NAACL)*, pages 966–974, Los Angeles, CA, USA, June. Association for Computational Linguistics.

Saša Hasan and Hermann Ney. 2009. Comparison of Extended Lexicon Models in Search and Rescoring for SMT. In *Proc. of the Human Language Technology Conf. / North American Chapter of the Association for Computational Linguistics (HLT-NAACL)*, volume short papers, pages 17–20, Boulder, CO, USA, June.

Saša Hasan, Juri Ganitkevitch, Hermann Ney, and Jesús Andrés-Ferrer. 2008. Triplet lexicon models for statistical machine translation. In *Proc. of the Conf. on Empirical Methods for Natural Language Processing (EMNLP)*, pages 372–381.

Liang Huang and David Chiang. 2007. Forest Rescoring: Faster Decoding with Integrated Language Models. In *Proc. of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 144–151, Prague, Czech Republic, June.

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

Gonzalo Iglesias, 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.

Kevin Knight. 1999. Decoding Complexity in Word-Replacement Translation Models. *Computational Linguistics*, 25(4):607–615, December.

Arne Mauser, 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.

Franz Josef Och and Hermann Ney. 2002. Discriminative Training and Maximum Entropy Models for Statistical Machine Translation. In *Proc. of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 295–302, Philadelphia, Pennsylvania, USA, July.

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

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

David Vilar, 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.

Richard Zens and Hermann Ney. 2008. Improvements in Dynamic Programming Beam Search for Phrase-Based Statistical Machine Translation. In *International Workshop on Spoken Language Translation*, pages 195–205, Honolulu, Hawaii, October.

Richard Zens, 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.