Jane 2: Open Source Phrase-based and Hierarchical Statistical Machine Translation

Joern Wuebker Matthias Huck Stephan Peitz Malte Nuhn
Markus Freitag Jan-Thorsten Peter Saab Mansour Hermann Ney
Human Language Technology and Pattern Recognition Group
Computer Science Department
RWTH Aachen University
D-52056 Aachen, Germany
<surname>@cs.rwth-aachen.de

ABSTRACT
We present Jane 2, an open source toolkit supporting both the phrase-based and the hierarchical phrase-based paradigm for statistical machine translation. It is implemented in C++ and provides efficient decoding algorithms and data structures. This work focuses on the description of its phrase-based functionality. In addition to the standard pipeline, including phrase extraction and parameter optimization, Jane 2 contains several state-of-the-art extensions and tools. Forced alignment phrase training can considerably reduce rule table size while learning the translation scores in a more principled manner. Word class language models can be used to integrate longer context with a reduced vocabulary size. Rule table interpolation is applicable for different tasks, e.g. domain adaptation. The decoder distinguishes between lexical and coverage pruning and applies reordering constraints for efficiency.

KEYWORDS: statistical machine translation, open source toolkit, phrase-based translation, hierarchical translation.

Proceedings of COLING 2012: Demonstration Papers, pages 483–492,
COLING 2012, Mumbai, December 2012.

1 Introduction

This work describes version 2 of Jane, an open source statistical machine translation (SMT) toolkit. Jane 2 provides implementations for the standard pipeline for SMT, including rule table generation, parameter optimization, and decoding. The two dominating paradigms in current research, the phrase-based (Koehn et al., 2003) and the hierarchical (Chiang, 2007) approach to SMT, are fully supported. While there are other open source toolkits available which are capable of performing similar or even the same tasks, Jane 2 has some unique properties that make it an attractive alternative for research.

Efficiency. Jane 2 implements several different decoding algorithms which make use of state-of-the-art pruning techniques and efficient data structures in order to minimize memory usage and runtime. It is capable of on-demand loading of language and translation models, and its flexible parameterization allows for fine-grained configuration tradeoffs between efficiency and translation quality.

Parallelization. Most operations—including phrase extraction, optimization, and decoding—can be parallelized under an Oracle Grid Engine or Platform LSF batch system.

Documentation. The extensive manual (Vilar et al., 2012b) contains simple walkthroughs to get started as well as descriptions of the features and their parameters.

Extensibility. A modular design and flexible extension mechanisms allow for easy integration of novel features and translation approaches.

Jane is developed in C++ with special attention to clean code. It was originally released as a purely hierarchical machine translation toolkit. Version 1 is described in detail in (Vilar et al., 2010a), (Stein et al., 2011), and (Vilar et al., 2012a). Jane 2 is available under an open source non-commercial license and can be downloaded from `www.hltpr.rwth-aachen.de/jane`. Here we focus on presenting Jane's phrase-based translation mode, which has been added to the toolkit in version 2.*

2 Related Work

Moses (Koehn et al., 2007) is a widely used open source toolkit for statistical machine translation. It was originally designed for phrase-based decoding, but now also supports the hierarchical paradigm. Moses provides tools for the complete machine translation pipeline, contains implementations for a wide variety of different models and is well documented.

Joshua (Li et al., 2009) is written in Java and implements the full pipeline for hierarchical machine translation. In addition to standard hierarchical rule tables, it is capable of extracting syntax augmented machine translation (SAMT) grammars (Zollmann and Venugopal, 2006).

cdec (Dyer et al., 2010) is a flexible decoding framework with a unified representation for translation forests.

Ncode (Crego et al., 2011) implements the *n*-gram-based approach to machine translation (Mariño et al., 2006). Reordering is performed by creating a lattice in a preprocessing step, which is passed on to the monotone decoder.

Phrasal (Cer et al., 2010) is an open source machine translation package with a Java implementation of the phrase-based machine translation paradigm. Phrasal is capable of extracting and translating with discontinuous phrases (Galley and Manning, 2010).

NiuTrans (Xiao et al., 2012) is developed in C++ and supports phrase-based, hierarchical phrase-based and syntax-based models.

*See (Huck et al., 2012b) for a description of novel features for hierarchical translation in version 2 of Jane.

3 Overview of the Jane 2 Open Source SMT Toolkit

3.1 Rule extraction

Jane 2 provides a single-command framework for rule extraction of both hierarchical and phrase-based rule tables. Rule extraction is done using a two pass algorithm which allows extracting only the rules needed to translate a specific corpus. This is especially useful for cutting down the large amount of rules that arise during extraction of hierarchical rules. A binary rule table format allows on-demand loading of the necessary phrases to minimize memory consumption. Both hierarchical and phrase-based extraction implement heuristics to make sure that every word is extracted with a single-word phrase, even if they are are not consistent with the bilingual alignment. Besides calculating source-to-target and target-to-source phrase probabilities, Jane 2 features a customizable IBM1 scorer and binary count features. Further, Jane 2 includes multiple tools that allow pruning, filtering and modifying rule tables.

In the standard setting, each sentence pair in the training corpus is assigned a weight of 1. A new feature in Jane 2 is weighted phrase extraction for phrase-based rules, which allows assigning arbitrary weights for each sentence pair. This feature can be utilized for domain adaptation, where the weight represents the relatedness of the sentence pair to the domain.

3.2 Rule table interpolation

Jane 2 also includes a functionality for rule table interpolation which is especially interesting for combining in-domain and out-of-domain data. Having specified a set of rule tables $T_1, \ldots, T_i, \ldots, T_I$ to interpolate, Jane 2 can be configured to include all combinations of union and intersection for the entries contained in the input rule tables. Furthermore, the number and types of features to create from the input tables can be specified. Currently available options include *loglinear* ($\sum_{i=1}^{I} f_i \cdot c_i$), *linear* ($\log \sum_{i=1}^{I} \exp(f_i) \cdot c_i$), *copy* ($f_i$, i fixed), *max* ($\max_{i=1}^{I} f_i$) and *ifelse* (f_i,lowest i s.t. T_i contains the rule). The algorithm to create the output table is efficient (linear time), given the input rule tables are sorted.

3.3 Decoders

Hierarchical translation. Jane implements three parsing-based search strategies for hierarchical translation: *cube pruning* (Chiang, 2007), *cube growing* (Huang and Chiang, 2007) with various heuristics for language model score computation (Vilar and Ney, 2009), and *source cardinality synchronous cube pruning* (Vilar and Ney, 2012). Pruning settings can be configured flexibly for all hierarchical search algorithms.

Phrase-based translation. The phrase-based decoding algorithm in Jane 2 is a *source cardinality synchronous search* (SCSS) procedure and applies separate pruning to lexical and coverage hypotheses similar to (Zens and Ney, 2008). The distinction between lexical and coverage hypotheses has been shown to have a significant positive effect on the scalability of the algorithm. For efficient decoding, language model look-ahead (Wuebker et al., 2012) can be applied. Jane 2 also provides an additional *FastSCSS* decoder, which can only produce single-best output, but is considerably faster by not maintaining separate model costs and by deleting recombined hypotheses.

3.4 Optimization

Log-linear feature weights (Och and Ney, 2002) can be optimized with either the Downhill Simplex algorithm (Nelder and Mead, 1965), Och's minimum error rate training (MERT) (Och, 2003), or the Margin Infused Relaxed Algorithm (MIRA) (Chiang et al., 2009).

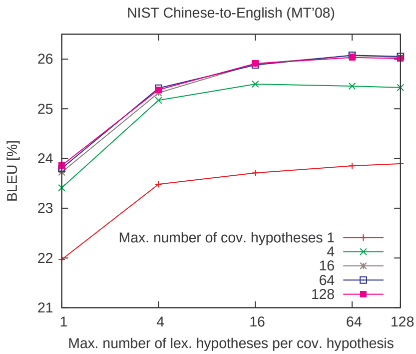

Figure 1: Effect of pruning parameters in the phrase-based decoder for the NIST Chinese→English translation task.

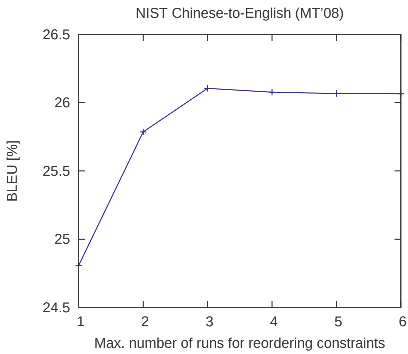

Figure 2: Effect of IBM phrase reordering constraints in the phrase-based decoder for the NIST Chinese→English translation task.

The challenge for optimization techniques is to find a good local optimum while avoiding bad local optima. Downhill Simplex and Och's method work well for a relatively small set of scaling factors. In experiments, Och's method yields better results and needs a lower number of iterations than Downhill Simplex. Both Downhill Simplex and Och's method have problems with large amounts of scaling factors (Chiang et al., 2008). (Watanabe et al., 2007) first used MIRA in SMT, which the authors claim to work well with a huge amount of features. (Chiang et al., 2009) get a significant improvement with an extremely large amount of features optimized by MIRA. Our implementation is very similar to the one presented in the above mentioned papers. MIRA is a good choice for a scaling factor set of more than 40 features.

3.5 Additional functionality

Jane additionally implements a number of advanced techniques. These range from discriminative word lexicon (DWL) models and triplet lexicon models (Mauser et al., 2009; Huck et al., 2010) over syntactic enhancements like parse matching (Vilar et al., 2008), preference grammars (Venugopal and Zollmann, 2009; Stein et al., 2010), soft string-to-dependency translation (Peter et al., 2011) and pseudo-syntactic enhancements like poor man's syntax (Vilar et al., 2010b) to discriminative lexicalized reordering extensions (Huck et al., 2012a).

4 Phrase-based Translation with Jane 2

4.1 Lexical and coverage pruning

In this section, we evaluate the effect of lexical pruning per coverage and coverage pruning per cardinality (Zens and Ney, 2008) in Jane's phrase-based decoder.

For a foreign input sentence f_1^J of length J, the set of source positions that are already translated (*covered*) in one state of the search process of the phrase-based translation system is called a *coverage* $C \subseteq \{1, \ldots, J\}$. *Lexical hypotheses* may differ in their coverage, in the current source sentence position, as well as in their language model history. The term *coverage hypothesis* is used to refer to the set of all lexical hypotheses with the same coverage C. In *lexical pruning per coverage*, the scores of all lexical hypotheses that have the same coverage C are compared. In *coverage pruning per cardinality*, the scores of all coverage hypotheses

	English→French		German→English	
	Bleu [%]	Ter [%]	Bleu [%]	Ter [%]
Baseline	31.7	50.5	29.2	50.2
+ word class LM	32.0	50.1	29.8	49.7

Table 1: Comparison of baseline systems and systems augmented with a 7-gram word class language model on different language pairs.

that share the same cardinality $c = |C|$ are compared. The score of a coverage hypothesis is for this purpose defined as the maximum score of any lexical hypothesis with coverage C. Histogram pruning is applied with parameters N_C for coverage pruning per cardinality and N_L for lexical pruning per coverage. Thus, if there are more than N_C coverage hypotheses for a particular cardinality c, only the best N_C candidates are kept, and if there are more than N_L lexical hypotheses for a particular coverage C, only the best N_L candidates are kept, respectively. Note that all lexical hypotheses with coverage C are dismissed if a coverage hypothesis C gets pruned.

We present empirical results on the NIST Chinese→English MT'08 translation task (NIST, 2008). We work with a parallel training corpus of 3.0 M Chinese–English sentences pairs (77.5 M Chinese / 81.0 M English running words). We evaluate all combinations of $N_C \in \{1, 4, 16, 64, 128\}$ and $N_L \in \{1, 4, 16, 64, 128\}$. The results are shown in Figure 1. Values beyond 16 of any of the two pruning parameters barely yield any additional improvement.

4.2 Reordering constraints

Restricting the possible reorderings is important in order to keep the search procedure tractable (Knight, 1999). Many decoders are limited to applying a jump distance limit. The search algorithm implemented in Jane 2 in addition is capable of discarding all source-side coverages with more than a maximum number of isolated contiguous runs. This restriction is known as *IBM phrase reordering constraints* (Zens et al., 2004). Configuring a maximum of one run is equivalent to monotone translation in this terminology. In the experiments from Section 4.1, we adopted the IBM phrase reordering constraints with a maximum of four runs and a jump distance limit of ten. We now evaluate the maximum runs parameter in the range from 1 to 6 with $N_C = 64$ and $N_L = 64$. The results are shown in Figure 2. Values beyond three do not improve translation quality any further, while monotone translation is considerably worse than translation with reorderings enabled.

4.3 Word class language models

In addition to the standard language model, a language model based on word classes can be used for phrase-based decoding in Jane 2. By clustering words into word classes, e.g. with the tool *mkcls* (Och, 2000), the vocabulary size is reduced and language models with higher n-gram order can be trained. By using a higher order in the translation process, the decoder is able to capture long-range dependencies.

In Table 1 the impact of the word class language model on different language pairs is shown. The experiments were carried out on the English→French and German→English MT tracks (TED task) of the IWSLT 2012 evaluation campaign (IWSLT, 2012). By applying a 7-gram word class language model, we achieve improvements of up to +0.6% Bleu and 0.5% Ter.

system	BLEU [%]	TER [%]	memory	words/sec
Jane	20.1	63.7	10G	7.7 (18.1)
Moses	19.0	65.1	22G	1.8
Moses with Jane rule table	20.1	63.8	19G	1.9

Table 2: Comparison of Moses with the phrase-based Jane 2 SCSS decoder, and its fast implementation optimized for single-best output (FastSCSS, in parentheses). All models are loaded into memory before decoding and loading time is eliminated for speed computation.

4.4 Forced alignment phrase training

Jane 2 features a framework to easily perform forced alignment phrase training, as described by (Wuebker et al., 2010). Phrase training is called with a single command for any number of iterations. Leave-one-out and cross-validation are automatically applied. It is made efficient by first performing bilingual phrase matching before search and by discarding the language model. To achieve good coverage of the training data, *backoff phrases* can be added to the translation model on-the-fly and *fallback runs* allow the decoder to retry with different parameterization, if aligning a sentence pair failed. This phrase training can considerably reduce rule table size, while providing a more statistcally sound way of estimating the translation probabilities.

4.5 Comparison with Moses

We compare the phrase-based decoder implemented in Jane 2 with Moses on the German→English task of the *EMNLP 2011 Sixth Workshop on Statistical Machine Translation* (WMT, 2011) on `newstest2009` in Table 2, keeping track of memory consumption and decoding speed. We use the same 4-gram LM for both Moses and Jane, and MERT is run separately for each setup. Jane's rule table is trained with three iterations of forced alignment (see Section 4.4). Moses is run in its standard setup (without lexicalized reordering models). For comparison we also ran Moses with our rule table. In this setup, Jane outperforms Moses by 1.1% BLEU. Moses can close the gap by using Jane's rule table. When the translation and language model are loaded into memory, Jane's memory consumption is about half that of Moses, and it is four times faster (ten times when using the FastSCSS decoder).

5 Conclusions

Jane is a flexible and efficient state-of-the-art SMT toolkit that is freely available to the scientific community. Jane's implementation of a source cardinality synchronous search algorithm for phrase-based translation has been released with version 2 of the toolkit. The algorithm applies separate pruning to lexical and coverage hypotheses and allows for restricting the possible reorderings via IBM phrase reordering constraints. A word class language model can be utilized during decoding. Phrase translation models can optionally be trained using forced alignment with leave-one-out.

Acknowledgments

This work was partly achieved as part of the Quaero Programme, funded by OSEO, French State agency for innovation. The research leading to these results has also received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement n° 287658 and the FP7 project T4ME Net, Contract n° 249119.

References

Cer, D., Galley, M., Jurafsky, D., and Manning, C. D. (2010). Phrasal: A Statistical Machine Translation Toolkit for Exploring New Model Features. In *Proceedings of the NAACL HLT 2010 Demonstration Session*, pages 9–12, Los Angeles, CA, USA.

Chiang, D. (2007). Hierarchical phrase-based translation. *Computational Linguistics*, 33(2):201–228.

Chiang, D., Knight, K., and Wang, W. (2009). 11,001 New Features for Statistical Machine Translation. In *Proc. of the Human Language Technology Conf. / North American Chapter of the Assoc. for Computational Linguistics (HLT-NAACL)*, pages 218–226, Boulder, CO, USA.

Chiang, D., Marton, Y., and Resnik, P. (2008). Online Large-Margin Training of Syntactic and Structural Translation Features. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 224–233, Honolulu, HI, USA.

Crego, J. M., Yvon, F., and Mariño, J. B. (2011). Ncode: an Open Source Bilingual N-gram SMT Toolkit. *The Prague Bulletin of Mathematical Linguistics*, 96:49–58.

Dyer, C., Lopez, A., Ganitkevitch, J., Weese, J., Ture, F., Blunsom, P., Setiawan, H., Eidelman, V., and Resnik, P. (2010). cdec: A Decoder, Alignment, and Learning framework for finite-state and context-free translation models. In *Proceedings of the ACL 2010 System Demonstrations*, ACLDemos '10, pages 7–12, Uppsala, Sweden.

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

Huang, L. and Chiang, D. (2007). Forest Rescoring: Faster Decoding with Integrated Language Models. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*, pages 144–151, Prague, Czech Republic.

Huck, M., Peitz, S., Freitag, M., and Ney, H. (2012a). Discriminative Reordering Extensions for Hierarchical Phrase-Based Machine Translation. In *Proc. of the 16th Annual Conf. of the European Association for Machine Translation (EAMT)*, pages 313–320, Trento, Italy.

Huck, M., Peter, J.-T., Freitag, M., Peitz, S., and Ney, H. (2012b). Hierarchical Phrase-Based Translation with Jane 2. *The Prague Bulletin of Mathematical Linguistics*, (98):37–50.

Huck, M., Ratajczak, M., Lehnen, P., and Ney, H. (2010). A Comparison of Various Types of Extended Lexicon Models for Statistical Machine Translation. In *Proc. of the Conf. of the Assoc. for Machine Translation in the Americas (AMTA)*, Denver, CO, USA.

IWSLT (2012). TED task of the IWSLT 2012 evaluation campaign. `http://www.iwslt2012.org/index.php/evaluation-campaign/ted-task/`.

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

Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantine, A., and Herbst, E. (2007). Moses: Open Source Toolkit for Statistical Machine Translation. In *Annual Meeting of the Association for Computational Linguistics (ACL), demonstration session*, pages 177–180, Prague, Czech Republic.

Koehn, P., Och, F. J., and Marcu, D. (2003). Statistical Phrase-Based Translation. In *Proceedings of the 2003 Meeting of the North American chapter of the Association for Computational Linguistics (NAACL-03)*, pages 127–133, Edmonton, Canada.

Li, Z., Callison-Burch, C., Dyer, C., Ganitkevitch, J., Khudanpur, S., Schwartz, L., Thornton, W., Weese, J., and Zaidan, O. (2009). Joshua: An Open Source Toolkit for Parsing-based Machine Translation. In *Proceedings of the 4th EACL Workshop on Statistical Machine Translation (WMT09)*, pages 135–139, Athens, Greece.

Mariño, J. B., Banchs, R. E., Crego, J. M., de Gispert, A., Lambert, P., Fonollosa, J. A. R., and Costa-Jussà, M. R. (2006). N-gram-based Machine Translation. *Computational Linguistics*, 32(4):527–549.

Mauser, A., Hasan, S., and Ney, H. (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 210–217, Singapore.

Nelder, J. A. and Mead, R. (1965). A Simplex Method for Function Minimization. *The Computer Journal*, 7:308–313.

NIST (2008). Open machine translation 2008 evaluation (MT08). `http://www.itl.nist.gov/iad/mig/tests/mt/2008/`.

Och, F. J. (2000). mkcls: Training of word classes. `http://www.hltpr.rwth-aachen.de/web/Software/mkcls.html`.

Och, F. J. (2003). Minimum Error Rate Training in Statistical Machine Translation. In *Proc. of the 41th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 160–167, Sapporo, Japan.

Och, F. J. and Ney, H. (2002). Discriminative Training and Maximum Entropy Models for Statistical Machine Translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 295–302, Philadelphia, PA, USA.

Peter, J.-T., Huck, M., Ney, H., and Stein, D. (2011). Soft String-to-Dependency Hierarchical Machine Translation. In *Proc. of the Int. Workshop on Spoken Language Translation (IWSLT)*, pages 246–253, San Francisco, CA, USA.

Stein, D., Peitz, S., Vilar, D., and Ney, H. (2010). A Cocktail of Deep Syntactic Features for Hierarchical Machine Translation. In *Proc. of the Conf. of the Assoc. for Machine Translation in the Americas (AMTA)*, Denver, CO, USA.

Stein, D., Vilar, D., Peitz, S., Freitag, M., Huck, M., and Ney, H. (2011). A Guide to Jane, an Open Source Hierarchical Translation Toolkit. *The Prague Bulletin of Mathematical Linguistics*, (95):5–18.

Venugopal, A. and Zollmann, A. (2009). Preference Grammars: Softening Syntactic Constraints to Improve Statistical Machine Translation. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 236–244, Boulder, Colorado, USA.

Vilar, D. and Ney, H. (2009). On LM Heuristics for the Cube Growing Algorithm. In *Proc. of the Annual Conf. of the European Assoc. for Machine Translation (EAMT)*, pages 242–249, Barcelona, Spain.

Vilar, D. and Ney, H. (2012). Cardinality pruning and language model heuristics for hierarchical phrase-based translation. *Machine Translation*, 26(3):217–254.

Vilar, D., Stein, D., Huck, M., and Ney, H. (2010a). 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.

Vilar, D., Stein, D., Huck, M., and Ney, H. (2012a). Jane: an advanced freely available hierarchical machine translation toolkit. *Machine Translation*, 26(3):197–216.

Vilar, D., Stein, D., Huck, M., Wuebker, J., Freitag, M., Peitz, S., Nuhn, M., and Peter, J.-T. (2012b). Jane: User's Manual. `http://www.hltpr.rwth-aachen.de/jane/manual.pdf`.

Vilar, D., Stein, D., and Ney, H. (2008). Analysing Soft Syntax Features and Heuristics for Hierarchical Phrase Based Machine Translation. In *Proc. of the Int. Workshop on Spoken Language Translation (IWSLT)*, pages 190–197, Waikiki, HI, USA.

Vilar, D., Stein, D., Peitz, S., and Ney, H. (2010b). If I Only Had a Parser: Poor Man's Syntax for Hierarchical Machine Translation. In *Proc. of the Int. Workshop on Spoken Language Translation (IWSLT)*, pages 345–352, Paris, France.

Watanabe, T., Suzuki, J., Tsukada, H., and Isozaki, H. (2007). Online Large-Margin Training for Statistical Machine Translation. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 764–773, Prague, Czech Republic.

WMT (2011). EMNLP 2011 Sixth Workshop on Statistical Machine Translation. `http://www.statmt.org/wmt11/`.

Wuebker, J., Mauser, A., and Ney, H. (2010). Training Phrase Translation Models with Leaving-One-Out. In *Proc. of the 48th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 475–484, Uppsala, Sweden.

Wuebker, J., Ney, H., and Zens, R. (2012). Fast and Scalable Decoding with Language Model Look-Ahead for Phrase-based Statistical Machine Translation. In *Annual Meeting of the Assoc. for Computational Linguistics*, pages 28–32, Jeju, Republic of Korea.

Xiao, T., Zhu, J., Zhang, H., and Li, Q. (2012). NiuTrans: An Open Source Toolkit for Phrase-based and Syntax-based Machine Translation. In *Proceedings of the ACL 2012 System Demonstrations*, pages 19–24, Jeju, Republic of Korea.

Zens, R. and Ney, H. (2008). Improvements in Dynamic Programming Beam Search for Phrase-based Statistical Machine Translation. In *Proc. of the Int. Workshop on Spoken Language Translation (IWSLT)*, pages 195–205, Honolulu, HI, USA.

Zens, R., Ney, H., Watanabe, T., and Sumita, E. (2004). Reordering Constraints for Phrase-Based Statistical Machine Translation. In *COLING '04: The 20th Int. Conf. on Computational Linguistics*, pages 205–211, Geneva, Switzerland.

Zollmann, A. and Venugopal, A. (2006). Syntax Augmented Machine Translation via Chart Parsing. In *Proc. of the Workshop on Statistical Machine Translation (WMT)*, pages 138–141, New York City, NY, USA.