Faster and Lighter Phrase-based Machine Translation Baseline

Liling Tan, Jon Dehdari and Josef van Genabith
Universität des Saarlandes, Germany
liling.tan@uni-saarland.de, jon.dehdari@dfki.de,
josef.van_genabith@dfki.de

Abstract

This paper describes the SENSE machine translation system participation in the Third Workshop for Asian Translation (WAT2016). We share our best practices to build a fast and light phrase-based machine translation (PBMT) models that have comparable results to the baseline systems provided by the organizers. As Neural Machine Translation (NMT) overtakes PBMT as the state-of-the-art, deep learning and new MT practitioners might not be familiar with the PBMT paradigm and we hope that this paper will help them build a PBMT baseline system quickly and easily.

1 Introduction

With the advent of Neural Machine Translation (NMT), the Phrased-Based Machine Translation (PBMT) paradigm casts towards the sunset (Neubig et al., 2015; Sennrich et al., 2016; Bentivogli et al., 2016; Wu et al., 2016; Crego et al., 2016). As the NMT era dawns, we hope to document the best practices in building a fast and light phrase-based machine translation baseline. In this paper, we briefly describe the PBMT components, list the tools available for PBMT systems prior to the neural tsunami, and present our procedures to build fast and light PBMT models with our system's results in the WAT2016 (Nakazawa et al., 2016).

1.1 Phrase-Based Machine Translation

The objective of the machine translation system is to find the best translation $\hat{\boldsymbol{t}}$ that maximizes the translation probability p($\boldsymbol{t}|\boldsymbol{s}$) given a source sentence $\boldsymbol{s}$; mathematically:

$$\hat{t} = \underset{t}{argmax}\, p(t|s) \tag{1}$$

Applying the Bayes' rule, we can factorized the p($\boldsymbol{t}|\boldsymbol{s}$) into three parts:

$$p(t|s) = \frac{p(t)}{p(s)} p(s|t) \tag{2}$$

Substituting our p($\boldsymbol{t}|\boldsymbol{s}$) back into our search for the best translation $\hat{\boldsymbol{t}}$ using *argmax*:

$$\begin{aligned} \hat{t} &= \underset{t}{argmax}\, p(t|s) \\ &= \underset{t}{argmax}\, \frac{p(t)}{p(s)} p(s|t) \\ &= \underset{t}{argmax}\, p(t)p(s|t) \end{aligned} \tag{3}$$

We note that the denominator p($\boldsymbol{s}$) can be dropped because for all translations the probability of the source sentence remains the same and the *argmax* objective optimizes the probability relative to the set of possible translations given a single source sentence. The p($\boldsymbol{t}|\boldsymbol{s}$) variable can be viewed as the bilingual dictionary with probabilities attached to each entry to the dictionary (*aka* **phrase table**). The p($\boldsymbol{t}$) variable

Proceedings of the 3rd Workshop on Asian Translation,
pages 184–193, Osaka, Japan, December 11-17 2016.

governs the grammaticality of the translation and we model it using an ***n*-gram language model** under the PBMT paradigm.

Machine Translation developed rapidly with the introduction of IBM **word alignment** models (Brown et al., 1990; Brown et al., 1993) and *word-based* MT systems performed word-for-word decoding word alignments and n-gram language model.

The word-based systems eventually developed into the phrase-based systems (Och and Ney, 2002; Marcu and Wong, 2002; Zens et al., 2002; Koehn et al., 2003) which relies on the word alignment to generate phrases. The phrase-based models translate contiguous sequences of words from the source sentence to contiguous words in the target language. In this case, the term *phrase* does not refer to the linguistic notion of syntactic constituent but the notion of n-grams. Knight (1999) defined the word/phrase-based model as a search problem that grows exponentially to the sentence length. The phrase-based models significantly improve on the word-based models, especially for closely-related languages. This mainly due to the modeling of local reordering and the assumption that most orderings of contiguous n-grams are monotonic. However, that is not the case of translation between language pairs with different syntactic constructions; e.g. when translating between SVO-SOV languages.

Tillmann (2004) and Al-Onaizan and Papineni (2006) proposed several **lexicalized reordering** and distortion models to surmount most long-distance reordering issues. Alternatively, to overcome reordering issues with simple distortion penalty, Zollmann et al. (2008) memorized a larger phrase n-grams sequence from a huge training data and allow larger distortion limits; it achieves similar results to more sophisticated reordering techniques with lesser training data. In practice, reordering is set to a small window and Birch et al. (2010) has shown that phrase-based models perform poorly even with short and medium range reordering.

Och and Ney (2002) simplified the integration of additional model components using the *log-linear model*. The model defines feature functions *h(x)* with weights λ in the following form:

$$P(x) = \frac{exp(\sum_{i=1}^{n} \lambda_i h_i(x))}{Z} \tag{4}$$

where the normalization constant *Z* turns the numerator into a probability distribution.

In the case of a simple model in Equation (3), it contains the two primary features, we define the components as such:

$$\begin{aligned} h_1(x) &= logp(t) \\ h_2(x) &= logp(s|t) \end{aligned} \tag{5}$$

where the *h(x*$_1$*)* and *h(x*$_2$*)* are associated with the λ_1 and λ_2 respectively.

The flexibility of the log-linear model allows for additional translation feature components to be added to the model easily, e.g. the lexicalized reordering is modeled as additional feature(s) *h(x*$_i$*)* in PBMT. Additionally, the weights λ associated with the ***n*** components can be tuned to optimize the translation quality over the parallel sentences, ***D*** (often known as the development set):

$$\lambda_1^n = \underset{\lambda_1^n}{argmax} \sum_{d=1}^{D} \log P_{\lambda_1^n}(t_d|s_d) \tag{6}$$

Minimum Error Rate Training (MERT), a co-ordinate descent learning algorithm, is one of the commonly used algorithms used for tuning the the λ weights.

The resulting PBMT system is generally made up of the following (i) n-gram language model(s), (ii) probabilistic phrase table (optionally with additional feature(s)), (iii) probabilistic lexicalized reordering table and (iv) a set of λ weights for their respective *h(x)*.

The hierarchical phrase-based machine translation (aka *hiero*) extends the phrase-based models notion of phrase from naive contiguous words to a sequence of words and sub-phrases (Chiang, 2005). Within the hiero model, translation rules make use of the standard phrases and the reordering of the subphrases. Such reordering can be expressed as a lexicalized *gappy* hierarchical rule using X_1 and X_2 as placeholders for the subphrases.

At the onset of SMT, the importance of linguistic information to translation was recognized by Brown et al. (1993):

> *But it is not our intention to ignore linguistics, neither to replace it. Rather, we hope to enfold it in the embrace of a secure probabilistic framework so that the two together may draw strength from one another and guide us to better natural language processing systems in general and to better machine translation systems in particular.*

Factored SMT embarked on the task of effectively incorporating linguistics information from taggers, parses and morphological analyzers into the machine translation pipeline. It is motivated by fact that (i) linguistics information provides a layer of disambiguation to the ambiguity of natural language, (ii) generalized translation of out-of-vocabulary (OOV) words to overcome sparsity of training data and (iii) replace arbitrary limits with linguistics constraints put in place in the decoding process too keep the search space tractable (Hoang and Lopez, 2009; Koehn et al., 2010; Hoang, 2011).

Among the numerous Machine Translation tools, the Moses Statistical Machine Translation system is the de facto tool for building various machine translation models (vanilla, hierarchical or factored PBMT). The Pharaoh system is its predecessor (Koehn, 2004). Other than the Moses system, the Joshua[1] (Weese et al., 2011), Jane[2] (Vilar et al., 2010), Phrasal[3] (Cer et al., 2010) and `cdec`[4] (Dyer et al., 2010) systems are viable alternatives to build statistical MT models.

2 Fast and Light PBMT Setup

We used the phrase-based SMT implemented in the Moses toolkit (Koehn et al., 2003; Koehn et al., 2007) with the following vanilla Moses experimental settings:

i. Language modeling is trained using KenLM using 5-grams, with modified Kneser-Ney smoothing (Heafield, 2011; Kneser and Ney, 1995; Chen and Goodman, 1998). The language model is quantized to reduce filesize and improve querying speed (Whittaker and Raj, 2001; Heafield et al., 2013)

ii. Clustercat word clusters (Dehdari et al., 2016b) with `MGIZA++` implementation of IBM word alignment model 4 with grow-diagonal-final-and heuristics for word alignment and phrase-extraction (Koehn et al., 2003; Och and Ney, 2003; Gao and Vogel, 2008)

iii. Bi-directional lexicalized reordering model that considers monotone, swap and discontinuous orientations (Koehn, 2005; Galley and Manning, 2008)

iv. To minimize the computing load on the translation model, we compressed the phrase-table and lexical reordering model using Phrase Rank Encoding (Junczys-Dowmunt, 2012)

v. Minimum Error Rate Training (MERT) (Och, 2003) to tune the decoding parameters

Differing from the baseline systems proposed by the WAT2016 organizers, we have used (a) trie language model with quantization in *Step i* (b) Clustercat with multi-threaded word aligments (`MGIZA++`) instead of `mkcls` (Och, 1995) with `GIZA++` in *Step ii* and (c) phrase table compression in *Step iv*.

Although MT practitioners can use Moses' *Experiment Management System* (Koehn, 2010) to build a PBMT baseline, the models might not be easily modifiable due to the pre-coded configurations. The configuration constraints could become particularly frustrating when the model becomes prohibitively huge with limited read-only and random access memory.

[1] joshua.incubator.apache.org

[2] http://www-i6.informatik.rwth-aachen.de/jane/

[3] http://nlp.stanford.edu/phrasal/

[4] https://github.com/redpony/cdec

2.1 Quantization and Binarization of Language Models

Heafield et al. (2013) compared KenLM's trie data structure against other n-gram language model toolkit. He empirically showed that it uses less memory than the smallest model produced by other tools that creates lossless models and it was faster than SRILM (Stolcke, 2002) that also uses a trie data structure.

The floating point non-positive log probabilities of the n-gram and its backoff penalty can be stored in the trie exactly using 31 and 32 bits[5] respectively. These floating point values can be quantized using q bits per probability and r bit per backoff to save memory at the expense of decreased accuracy. KenLM uses the binning method to sort floats, divides them into equal size bins and averages the value within each bin. As such floats under the same bin shares the same value.

While quantization is lossy, we can use point compression (Whittaker and Raj, 2001) to remove the leading bits of the pointers and implicitly store the table of offsets into the array. Although point compression reduces the memory size of the language model, retrieving the offsets takes additional time.

The trie is produced by using the KenLM's `build_binary` tool. The quantization and trie binarization is performed using the last command below:

```
LM_ARPA=`pwd`/${TRAINING_DIR}/lm/lm.${LANG_E}.arpa.gz
LM_FILE=`pwd`/${TRAINING_DIR}/lm/lm.${LANG_E}.kenlm

${MOSES_BIN_DIR}/lmplz --order ${LM_ORDER} -S 80% -T /tmp \
< ${CORPUS_LM}.${LANG_E} | gzip > ${LM_ARPA}

${MOSES_BIN_DIR}/build_binary trie -a 22 -b 8 -q 8 ${LM_ARPA} ${LM_FILE}
```

The `-a` option sets the maximum number of leading bits that the point compression removes. The `-q` and `-b` options sets the number of bits to store the n-gram log probability and backoff respectively[6]. We can stack the point compression with quantization as shown above, the `-a 22 -b 8 -q 8` will set the maximum leading bits removal to 22 and stores the floating points for log probabilities and backoff penalties using 8 bits.

2.2 `MGIZA++` and Clustercat

Gao and Vogel (2008) implemented two parallelized versions of the original `GIZA++` tool, `PGIZA++` that uses multiple aligning processes where when the processes are finished, the master process collects the normalized counts and updates the model and child processes are restarted in the next iteration and `MGIZA++` that uses multi-threading on shared memory with locking mechanism to synchronize memory access.

Given a computing cluster (i.e. multiple machines), using `PGIZA++` would be appropriate whereas `MGIZA++` is suited for a single machine with multiple cores. An up-to-date fork of `MGIZA++` is maintained by the Moses community at `https://github.com/moses-smt/mgiza`.

While one might face issues with creating the `MGIZA++` binaries from source compilation[7], the Moses community provides pre-built binaries[8] on `http://www.statmt.org/moses/?n=moses.releases`. These can be easily downloaded and saved to a directory (e.g. `/path/to/moses-training-tools`) on the terminal as such:

```
wget -r -nH -nd -np -R index.html* \
http://www.statmt.org/moses/RELEASE-3.0/binaries/linux-64bit/training-tools/ \
-P /path/to/moses-training-tools
```

And the `EXT_BIN_DIR` variable in the training script can be set and be used in the translation model training process as such:

[5] Backoff penalty may sometimes be positive

[6] Note that unigram probabilities are never quantized

[7] Following the instructions on http://www.statmt.org/moses/?n=Moses.ExternalTools#ntoc3

[8] E.g. the direct link for the Linux OS can be found on http://www.statmt.org/moses/RELEASE-3.0/binaries/linux-64bit/training-tools/

```
EXT_BIN_DIR=/path/to/moses-training-tools/

${MOSES_SCRIPT}/training/train-model-10c.perl \
  --root-dir `pwd`/${TRAINING_DIR} \
  --model-dir `pwd`/${MODEL_DIR} \
  --corpus ${CORPUS} \
  --external-bin-dir ${EXT_BIN_DIR} \
  --mgiza -mgiza-cpus 10 \
  --f ${LANG_F} \
  --e ${LANG_E} \
  --parallel \
  --alignment grow-diag-final-and \
  --reordering msd-bidirectional-fe \
  --score-options "--GoodTuring" \
  --lm 0:${LM_ORDER}:${LM_FILE}:8 \
  --cores ${JOBS} \
  --sort-buffer-size 10G \
  --parallel \
  >& ${TRAINING_DIR}/training_TM.log
```

The `--mgiza` option activates the `MGIZA++` binary and `-mgiza-cpus 10` specifies the training to be done with 10 CPU threads. The default option is to use IBM model 4 where the probability for each word is conditioned on both the previously aligned word and on the word classes of its context words[9].

To generate the word classes, `MGIZA++` uses a single-threaded version of an old exchange clustering algorithm implementation, `mkcls`, which can be rather slow when the training corpus is sufficiently huge. Instead, we suggest the use of `Clustercat`[10], another exchange clustering algorithm that has a wrapper to emulate `mkcls` command-line interface and outputs. `Clustercat` is an implementation of the Bidirectional, Interpolated, Refining, and Alternating (BIRA) predictive exchange algorithm; notably, `ClusterCat` clusters a 1 billion token English News Crawl corpus in 1.5 hours while `mkcls` might take 3 days on the same machine (Dehdari et al., 2016a). To use `Clustercat` with `MGIZA++`, simply create a symbolic link the `mkcls` wrapper from `Clustercat` to the `moses-training-tools` directory, e.g.:

```
EXT_BIN_DIR=/path/to/moses-training-tools/
mv ${EXT_BIN_DIR}/mkcls mkcls-original
ln -s /path/to/clustercat/bin/mkcls ${EXT_BIN_DIR}/mkcls
```

2.3 Phrase Table and Lexicalized Reordering Table Compression

Extending the classic dictionary-based compression methods, Junczys-Dowmunt (2012) proposed the phrasal rank encoding compression algorithm where repeated sub-phrases are replaced by pointers in the phrase dictionary which results in a reduction in phrase table size. At decompression, the sub-phrases are looked up and re-inserted based on the pointers.

Strangely, Moses implementation of MERT releases the phrase table and lexicalized reordering tables after every cycle and reload it when attempting to decode the development data with the updated feature parameters. A reduced phrase table size would not only speed up the table loading in decoding time but more importantly, it speeds up the table loading at every MERT epoch.

The table compression tools are found in the Moses binary directory and can be activated while filtering the phrase table and lexicalized reordering table using `-Binarizer` option as shown below:

```
${MOSES_SCRIPT}/training/filter-model-given-input.pl \
  ${MODEL_DIR}.filtered/dev \
  ${MODEL_DIR}/moses.ini \
  ${DEV_F} \
  -Binarizer ${MOSES_BIN_DIR}/processPhraseTableMin ${MOSES_BIN_DIR}/processLexicalTableMin \
  -threads ${JOBS}
```

[9] `--giza-option` allows users to use train with other word alignment models

[10] https://github.com/jonsafari/clustercat

3 Results

Team	**Other Resources**	**System**	**BLEU**	**HUMAN**
JAPIO	JAPIO corpus	PBMT with pre-ordering	**58.66**	46.25
NTT	-	NMT with bidi-LSTM	44.99	**46.50**
NTT	-	PBMT with pre-ordering	40.75	39.25
SENSE	-	Vanilla PBMT (clustercat)	38.90	-
SENSE	-	Vanilla PBMT (mkcls)	38.75	-
ORGANIZER	-	Baseline PBMT	38.34	0

Table 1: Top Systems and Our Submissions to WAT2016 Patent Task (Chinese-Japanese)

Team	**Other Resources**	**System**	**BLEU**	**HUMAN**
NICT-2	ASPEC	PBMT with Preordering + Domain Adaptation	**34.64**	**14.00**
NICT-2	-	PBMT with Preordering + Domain Adaptation	34.64	-11.00
BJTU_NLP	-	NMT using RNN Encoder-Decoder with attention	32.79	-1.00
SENSE	-	Vanilla PBMT (clustercat)	32.11	-
ORGANIZER	-	Baseline PBMT	32.03	0
SENSE	-	Vanilla PBMT (mkcls)	31.84	-

Table 2: Top Systems and Our Submissions to WAT2016 Patent Task (Japanese-Chinese)

Using the fast and light PBMT system described in the previous section, we submitted the system outputs to the WAT 2016 shared task (Nakazawa et al., 2016) for Japanese to Chinese patent translation task and the Indonesian to English news domain task[11].

The Japan Patent Office (JPO) Patent corpus is the official resource provided for the Japanese-Chinese-Korean-English shared task. The training dataset is made up of 1 million sentences (250k each from the chemistry, electricity, mechanical engineering and physics domains). The Badan Pengkajian dan Penerapan Teknologi (BPPT) corpus is the official resource provided for the English-Indonesian shared task. The training dataset is made up of 1 million 50,000 training sentences from the general news domain.

Table 1 and 2 present our submission to the Japanese-Chinese Patent Task in WAT2016. Due to time constraint, we were not able to make the submission in time for the manual evaluation. Looking at the BLEU scores, we achieved relatively close BLEU scores for both translation directions as compared to the organizers' PBMT baseline.

From Table 1, we see that the NMT system achieved the best HUMAN score given a lower BLEU[12], this reinforced the rise of NMT era. More importantly, we see a huge difference in JAPIO's PBMT BLEU score (58.66) and NTT's NMT BLEU score (58.66) but both system achieved similar HUMAN scores. The same disparity in BLEU and HUMAN scores is evident from Table 2 where both NICT-2 PBMT systems (one trained with additional ASPEC corpus and the other without) scored 34.64 BLEU but the HUMAN score disparity ranges from -11.00 to +14.00. Such disparity reiterated the disparity between n-gram based metric and human evaluation in Tan et al. (2015a).

[11] In previous editions of WAT (Nakazawa et al., 2014; Nakazawa et al., 2015), we had participated using similar PBMT system in the English-Japanese-Chinese scientific text translation task using the ASPEC corpus, our results had been presented in Tan and Bond (2014) and Tan et al. (2015b) and in the Korean-English patent translation task using the JPO corpus (Tan et al., 2015a)

[12] Reported BLEU scores on JUMAN tokenizer

Team	System	BLEU	HUMAN
SENSE	Vanilla PBMT (clustercat)	**25.31**	1.250
SENSE	Vanilla PBMT (mkcls)	25.16	-2.750
ORGRANIZER	Online A	24.20	**35.75**
ORGRANIZER	Baseline PBMT	23.95	0
IITB	Bilingual Neural LM	22.35	-9.250
ORGRANIZER	Online B	18.09	10.50

Table 3: Results of WAT2016 English-Indonesian News Domain Task

Team	System	BLEU	HUMAN
ORGANIZER	Online A	**28.11**	**49.25**
SENSE	Vanilla PBMT (clustercat)	25.97	-8.25
SENSE	Vanilla PBMT (mkcls)	25.62	-5.00
ORGANIZER	Baseline PBMT	24.57	0
IITB	Bilingual Neural LM	22.58	-
ORGANIZER	Online B	19.69	34.50

Table 4: Results of WAT2016 Indonesian-English News Domain Task

Table 3 and 4 presents the results for the Indonesian-English News Domain Task. From Table 3, we achieve the highest BLEU scores in the English-Indonesia direction with a difference of >1.0+ BLEU score with respect to the baseline PBMT provided by the organizers. However, our HUMAN scores show that the quality of our system output is only marginally better than the baseline. Comparatively, the online system A has similar BLEU scores to the organizer's baseline but achieved stellar HUMAN scores of +35.75. Table 4 shows the results for the English-Indonesian task, the online system A and B achieved the best HUMAN scores. In both directions, we see the same automatic *vs* manual evaluation disparity from System B's low BLEU and high HUMAN scores and from our system's high BLEU and low/marginal HUMAN scores.

4 Conclusion

We motivate and describe the steps to build a fast and light phrase-based machine translation model that achieved comparable results to the WAT2016 baseline. We hope that our baseline system helps new MT practitioners that are not familiar with the Moses ecology[13] to build PBMT models. The full training script is available on `https://github.com/alvations/vanilla-moses/blob/master/train-vanilla-model.sh`.

Acknowledgements

The research leading to these results has received funding from the People Programme (Marie Curie Actions) of the European Union's Seventh Framework Programme FP7/2007-2013/ under REA grant agreement no. 317471.

References

Yaser Al-Onaizan and Kishore Papineni. 2006. Distortion models for statistical machine translation. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 529–536.

Luisa Bentivogli, Arianna Bisazza, Mauro Cettolo, and Marcello Federico. 2016. Neural versus Phrase-Based Machine Translation Quality: a Case Study. *ArXiv e-prints*, August.

Alexandra Birch, Miles Osborne, and Phil Blunsom. 2010. Metrics for mt evaluation: evaluating reordering. *Machine Translation*, 24(1):15–26.

[13] Our Vanilla PBMT system is complimentary to the steps described in http://www.statmt.org/moses/?n=Moses.Tutorial

Peter F Brown, John Cocke, Stephen A Della Pietra, Vincent J Della Pietra, Fredrick Jelinek, John D Lafferty, Robert L Mercer, and Paul S Roossin. 1990. A statistical approach to machine translation. *Computational linguistics*, 16(2):79–85.

Peter F Brown, Vincent J Della Pietra, Stephen A Della Pietra, and Robert L Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational linguistics*, 19(2):263–311.

Daniel Cer, Michel Galley, Daniel Jurafsky, and Christopher D Manning. 2010. Phrasal: a toolkit for statistical machine translation with facilities for extraction and incorporation of arbitrary model features. In *Proceedings of the NAACL HLT 2010 Demonstration Session*, pages 9–12.

Stanley Chen and Joshua T. Goodman. 1998. An empirical study of smoothing techniques for language modeling. Technical Report 10-98, Harvard University.

David Chiang. 2005. A hierarchical phrase-based model for statistical machine translation. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 263–270. Association for Computational Linguistics.

Josep Crego, Jungi Kim, Guillaume Klein, Anabel Rebollo, Kathy Yang, Jean Senellart, Egor Akhanov, Patrice Brunelle, Aurelien Coquard, Yongchao Deng, Satoshi Enoue, Chiyo Geiss, Joshua Johanson, Ardas Khalsa, Raoum Khiari, Byeongil Ko, Catherine Kobus, Jean Lorieux, Leidiana Martins, Dang-Chuan Nguyen, Alexandra Priori, Thomas Riccardi, Natalia Segal, Christophe Servan, Cyril Tiquet, Bo Wang, Jin Yang, Dakun Zhang, Jing Zhou, and Peter Zoldan. 2016. SYSTRAN's Pure Neural Machine Translation Systems. *ArXiv e-prints*, October.

Jon Dehdari, Liling Tan, and Josef van Genabith. 2016a. Bira: Improved predictive exchange word clustering. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1169–1174, San Diego, California, June. Association for Computational Linguistics.

Jon Dehdari, Liling Tan, and Josef van Genabith. 2016b. Scaling up word clustering. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, pages 42–46, San Diego, California, June. Association for Computational Linguistics.

Chris Dyer, Adam Lopez, Juri Ganitkevitch, Johnathan Weese, Ferhan Ture, Phil Blunsom, Hendra Setiawan, Vladimir Eidelman, and Philip Resnik. 2010. cdec: A decoder, alignment, and learning framework for finite-state and context-free translation models. In *Proceedings of the Association for Computational Linguistics (ACL)*.

Michel Galley and Christopher D Manning. 2008. A simple and effective hierarchical phrase reordering model. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 848–856.

Qin Gao and Stephan Vogel. 2008. Parallel implementations of word alignment tool. In *Software Engineering, Testing, and Quality Assurance for Natural Language Processing*, pages 49–57.

Kenneth Heafield, Ivan Pouzyrevsky, Jonathan H. Clark, and Philipp Koehn. 2013. Scalable modified Kneser-Ney language model estimation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 690–696, Sofia, Bulgaria.

Kenneth Heafield. 2011. KenLM: Faster and smaller language model queries. In *Proceedings of the EMNLP 2011 Sixth Workshop on Statistical Machine Translation*, pages 187–197, Edinburgh, Scotland, United Kingdom.

Hieu Hoang and Adam Lopez. 2009. A unified framework for phrase-based, hierarchical, and syntax-based statistical machine translation. In *In Proceedings of the International Workshop on Spoken Language Translation (IWSLT*, pages 152–159.

Hieu Hoang. 2011. *Improving statistical machine translation with linguistic information*. The University of Edinburgh.

Marcin Junczys-Dowmunt. 2012. Phrasal rank-encoding: Exploiting phrase redundancy and translational relations for phrase table compression. *The Prague Bulletin of Mathematical Linguistics*, 98:63–74.

Reinhard Kneser and Hermann Ney. 1995. Improved backing-off for m-gram language modeling. In *Acoustics, Speech, and Signal Processing, 1995. ICASSP-95., 1995 International Conference on*, volume 1, pages 181–184. IEEE.

Kevin Knight. 1999. Decoding complexity in word-replacement translation models. *Computational Linguistics*, 25(4):607–615.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 48–54.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, pages 177–180.

Philipp Koehn, Barry Haddow, Philip Williams, and Hieu Hoang. 2010. More linguistic annotation for statistical machine translation. In *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*, WMT '10, pages 115–120, Stroudsburg, PA, USA. Association for Computational Linguistics.

Philipp Koehn. 2004. Pharaoh: a beam search decoder for phrase-based statistical machine translation models. In *Machine translation: From real users to research*, pages 115–124. Springer.

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *Proceedings of MT Summit*, volume 5, pages 79–86.

Philipp Koehn. 2010. An experimental management system. *The Prague Bulletin of Mathematical Linguistics*, 94:87–96.

Daniel Marcu and William Wong. 2002. A phrase-based, joint probability model for statistical machine translation. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, pages 133–139. Association for Computational Linguistics.

Toshiaki Nakazawa, Hideya Mino, Isao Goto, Sadao Kurohashi, and Eiichiro Sumita. 2014. Overview of the first workshop on Asian translation. In *Proceedings of the First Workshop on Asian1 Translation (WAT2014)*, Tokyo, Japan.

Toshiaki Nakazawa, Hideya Mino, Isao Goto, Graham Neubig, Sadao Kurohashi, and Eiichiro Sumita. 2015. Overview of the 2nd workshop on Asian translation. In *Proceedings of the 2nd Workshop on Asian Translation (WAT2015)*, Kyoto, Japan.

Toshiaki Nakazawa, Hideya Mino, Isao Goto, Graham Neubig, Sadao Kurohashi, and Eiichiro Sumita. 2016. Overview of the 3rd workshop on Asian translation. In *Proceedings of the 3rd Workshop on Asian Translation (WAT2016)*, Kyoto, Japan.

Graham Neubig, Makoto Morishita, and Satoshi Nakamura. 2015. Neural Reranking Improves Subjective Quality of Machine Translation: NAIST at WAT2015. In *Proceedings of the 2nd Workshop on Asian Translation (WAT2015)*, pages 35–41, Kyoto, Japan, October.

Franz Josef Och and Hermann Ney. 2002. Discriminative training and maximum entropy models for statistical machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, July 6-12, 2002, Philadelphia, PA, USA.*, pages 295–302.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational linguistics*, 29(1):19–51.

Franz Och. 1995. Maximum-Likelihood-Schätzung von Wortkategorien mit Verfahren der kombinatorischen Optimierung. Bachelor's thesis (Studienarbeit), Universität Erlangen-Nürnburg.

Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*, pages 160–167.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Edinburgh neural machine translation systems for wmt 16. In *Proceedings of the First Conference on Machine Translation*, pages 371–376, Berlin, Germany, August. Association for Computational Linguistics.

Andreas Stolcke. 2002. Srilm-an extensible language modeling toolkit. In *Seventh International Conference on Spoken Language Processing*, pages 901–904.

Liling Tan and Francis Bond. 2014. Manipulating input data in machine translation. In *Proceedings of the 1st Workshop on Asian Translation (WAT2014)*.

Liling Tan, Jon Dehdari, and Josef van Genabith. 2015a. An Awkward Disparity between BLEU / RIBES Scores and Human Judgements in Machine Translation. In *Proceedings of the 2nd Workshop on Asian Translation (WAT2015)*, pages 74–81, Kyoto, Japan, October.

Liling Tan, Josef van Genabith, and Francis Bond. 2015b. Passive and pervasive use of bilingual dictionary in statistical machine translation. In *Proceedings of the Fourth Workshop on Hybrid Approaches to Translation (HyTra)*, pages 30–34, Beijing.

Christoph Tillmann. 2004. A unigram orientation model for statistical machine translation. In *Proceedings of HLT-NAACL 2004: Short Papers*, pages 101–104.

David Vilar, Daniel Stein, Matthias Huck, and Hermann Ney. 2010. Jane: Open source hierarchical translation, extended with reordering and lexicon models. In *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*, pages 262–270.

Jonathan Weese, Juri Ganitkevitch, Chris Callison-Burch, Matt Post, and Adam Lopez. 2011. Joshua 3.0: Syntax-based machine translation with the thrax grammar extractor. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 478–484.

Edward W. D. Whittaker and Bhiksha Raj. 2001. Quantization-based language model compression. In *Proceedings of INTERSPEECH*, pages 33–36.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, ukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation. *ArXiv e-prints*, September.

Richard Zens, Franz Josef Och, and Hermann Ney. 2002. Phrase-based statistical machine translation. In *KI 2002: Advances in Artificial Intelligence*, pages 18–32. Springer.

Andreas Zollmann, Ashish Venugopal, Franz Och, and Jay Ponte. 2008. A systematic comparison of phrase-based, hierarchical and syntax-augmented statistical mt. In *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*, pages 1145–1152.