I^{2}R Chinese-English Translation System for OpenMT 2015

Xuancong Wang, Cong Duy Vu Hoang, Kui Wu, Nina Zhou, Boon Hong Yeo, AiTi Aw, Haizhou Li
Department of Human Language Technology
Institute for Infocomm Research, Singapore
{wangxc, cdvhoang, wukui, zhoun, bhyeo, aaiti, hli}@i2r.a-star.edu.sg

Abstract

In this paper, we describe our system and approach used for the NIST Open Machine Translation 2015 (OpenMT15) evaluation campaign in the Chinese-to-English SMS/Chat and CTS category. A multi-pass approach was exploited to generate and select the best translation. First, we train multiple systems using the Moses phrase-based decoder and the Moses hierarchical-phrase-based decoder. Each system uses different features and pre/post-processing to ensure translation knowledge diversity. Next, for each of the system, we perform N-best rescoring by adding additional features to obtain 1-best translation. Finally, we combine the 1-best translation from each of these systems to select the best translation by re-scoring and re-ranking them with additional feature functions. In particular, this paper reports our effort in data processing, system training and system combination, as well as our performance on OpenMT15 Chinese-English task.

1 Introduction

This paper describes the statistical machine translation (SMT) system and approach undertaken by Institute for Infocomm Research (I^{2}R) for the NIST Open Machine Translation 2015 evaluation. We submitted runs under the constrained training condition for Chinese-to-English SMS/Chat and CTS domain.

A typical state-of-the-art SMT system consists of two passes (Koehn et al., 2003; Federico and Bertoldi, 2005). In the first pass, the N-best translations are generated by the decoding algorithm; in the second pass, the best translation is computed by re-scoring and re-ranking the N-best translations with additional feature functions.

For this task, we use a third pass to combine the output from individual systems to take advantage of the different features used in the different stages. For our system, in the first pass, we have trained several systems using the Moses phrase-based decoder and the Moses hierarchical-phrase-based decoder (Koehn et al., 2007). These systems use slightly different preprocessing, training data, alignment, features, etc., to ensure translation knowledge diversity. Features that applies to partial translation hypotheses are used. In the second pass, for each of the system, we perform N-best rescoring by adding additional features to obtain 1-best translation. These additional features typically only apply to complete translation hypotheses, so they cannot be used during on-the-fly decoding. In the third pass, we perform system combination from the 1-best hypothesis from different systems. Only model-independent features are used since the hypothesis from different systems may have different model-dependent features.

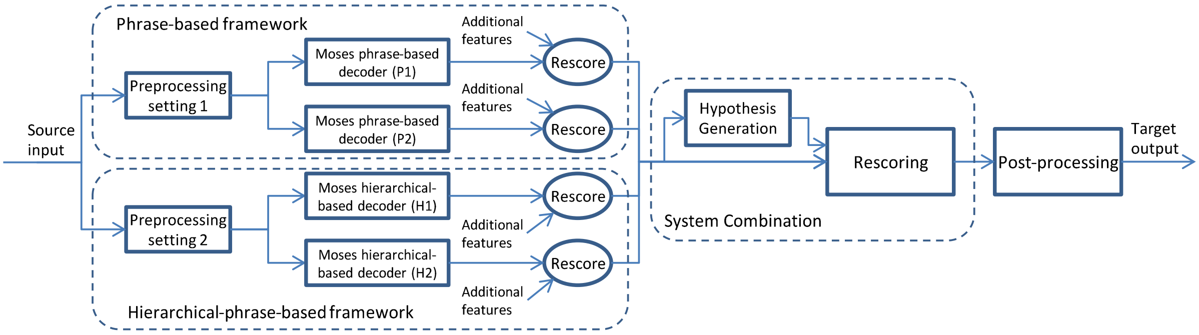

Figure 1: Overall system structure.

This paper is organized as follows. Section 2 presents the full structure of our system in details. Section 3 provides the experimental setups, reports and discusses the results obtained. It also describes the difference in our contrastive systems. Section 4 concludes this paper by giving a summary of our most useful findings.

2 System Description

The overall system architecture is shown in Figure 1. An input source sentence is preprocessed with two different settings, one for hierarchical-phrase-based framework, the other for phrase-based framework. Under each framework, we have trained multiple systems, all using Moses. For each system, the decoder produces N-best output. Additional features are added to perform rescoring to produce 1-best output for each system. After that, system combination is performed by hypothesis generation followed by N-best rescoring using a different set of global features.

2.1 Data Organization

Since this evaluation campaign is quite domain-specific, blindly adding out-of-domain data may degrade the performance. On the other hand, out-of-domain data can potentially increase the translation knowledge diversity which will improve the performance. For this task, different systems are trained using slightly different data selection to maximize translation knowledge diversity. This section will establish a common data set naming convention throughout this paper.

We have organized the data available under LDC data license agreement into different groups. The same grouping of data is used for all model trainings, but with different combinations and subsets of sentences.

Subgroup	Training data in LDC category
SMS-CHT	LDC2013E81 + LDC2013E85 + LDC2013E118 + LDC2013E125 + LDC2013E132
CTS	LDC2014E08 + LDC2014E50 + LDC2014E99
BC	LDC2009T02 + LDC2009T06 + LDC2013T11 + LDC2013T16 + LDC2007E101

Table 1: Corpus grouping for the relevant domain (Group 1)

In the rest of this paper, for easy understanding, we will refer the different datasets as in-domain, sub-general-domain, general-domain, out-domain, tuning, testing. The dataset grouping and corpus statistics are shown in Table 3.

2.2 Phrase-based Framework

We have built several phrase-based systems. Only the top 2 systems are selected for system combination. Table 4 shows the list of features used for one typical system.

Group	Dataset for *SMS-CHT*		Dataset for *CTS*	
	Name	**Approx. # of sent.**	**Name**	**Approx. # of sent.**
in-domain	*SMS-CHT*	130k	*CTS*	65k
subgeneral-domain	*SMS-CHT-CTS*	200k	*SMS-CHT-CTS*	200k
general-domain	*SMS-CHT-CTS-BC*	280k	*SMS-CHT-CTS-BC*	280k
out-domain	*NNB* + *PRH*	2M (from 8M)	*NNB* + *PRH*	2M (from 8M)
Tuning set	LDC2013E80 + LDC2013E83	6214	LDC2014E69	6636
Testing set	LDC2013E119	641	LDC2014E111	774

Table 3: Dataset grouping and corpus statistics.

Subgroup	**Training data in LDC category**
NNB	LDC2002E18.reduced + LDC2004T08.HK_News + LDC2005T06 + LDC2005T10 + LDC2007T09 + LDC2003E14 + LDC2008E56 + LDC2009E16 + LDC2009E95 + LDC2007E101 + LDC2010T03 + LDC2009T15 + LDC2008T18 + LDC2008T08 + LDC2007T23 + LDC2014T11 + LDC2014T04 + LDC2010T10 + LDC2010T11 + LDC2010T12 + LDC2010T14 + LDC2010T17 + LDC2010T21 + LDC2013T03 + LDC2013T07 + LDC2002T01 + LDC2004T07 + LDC2003T17 + LDC2006T04
PRH	LDC2004T08.HK_Hansards + LDC2004E12
LEX	LDC2002L27 + LDC2005T34

Table 2: Corpus grouping for the non-relevant domain (Group 2)

2.3 Preprocessing

The English text is tokenized using *don't* → *don 't* conventions. All the Chinese text is segmented using I^{2}R News Word Segmentation Engine. On the evaluation data, utterances are split according to sentence-end punctuation marks, translated by the system, and merged back.

2.4 Translation Model

To maximize translation knowledge diversity, we use GIZA++ (Och and Ney, 2000) and fast_align (Dyer et al., 2013) for performing word alignment, and concatenate all resulting alignments for the phrase extraction. In addition to this, we have combined 4 models (in-domain, subgeneral domain, general domain, out-domain) using perplexity minimization method in (Sennrich, 2012). For decoding, we use Moses phrase-based decoder for decoding. The decoder is tuned using batch MIRA algorithm to optimize the BLEU score on the tuning set.

2.5 Reordering

We have used two types of lexicalized reordering models developed by (Koehn et al., 2005) and (Galley and Manning, 2008). The former one measures the orientation of two phrases based on word alignments whereas the later one allows combinations of several phrases to determine the orientation. We also used the Operation Sequence Model (OSM) for joint translation and reordering (Durrani et al., 2011).

2.6 Rescoring

We use RNNLM and extended NNJM (Devlin et al., 2014) with bidirectional direction for rescoring. We re-adjust the weights of conventional n-gram language models (with data groups) with in-domain RNNLM and extended NNJM models.

Feature	Description
Translation Model (4 features)	The 4 standard feature: e2f, f2e, lex-e2f, lex-f2e TM-combine 4 models: in-domain, subgeneral-domain, general-domain, out-domain TM
Language Models (3 features)	in-domain 5-gram LM general-domain 5-gram LM out-domain 5-gram LM
Reordering Models 25 features	6 for lexicalized reordering with subgeneral domain, 8 for hierarchical reordering with subgeneral domain, 1 for distance-based distortion, 5 for OSM with in-domain, 5 for OSM with general-domain
Others (3 features)	UnknownWordPenalty, WordPenalty, PhrasePenalty
Rescoring	in-domain forward RNNLM in-domain S2T bidirectional NNJM
P2 SMS-CHT (wrt. P1-SMS-CHT)	- rescoring + in-domain S2T NNJM
P1 CTS (wrt. P1-SMS-CHT)	- in-domain 5-gram LM + subgeneral-domain 5-gram LM
P2 CTS (wrt. P1-CTS)	- rescoring + in-domain S2T NNJM + in-domain NPLM

Table 4: Features for the P1 system on SMS-CHT.

2.7 Hierarchical-phrase-based Framework

Our hierarchical-phrase-based framework consists of several systems. Each system does preprocessing, decoding and rescoring in cascade. Table 5 shows the list of features used for one typical system.

2.7.1 Preprocessing

The English text is tokenized using *don’t* → *do n’t* convention and lower-cased. The Chinese text is converted to half-width and simplified Chinese. Out-domain text is segmented using I^{2}R News Word Segmentation Engine. Other text is segmented using I^{2}R Informal Text Word Segmentation Engine. Punctuation correction is applied on the source text to pre-process SMS/Chat training, tuning, testing, and evaluation text. On the evaluation data, utterances are split according to the corrected sentence-end punctuation marks, translated by the system, and merged back.

Feature	Description
Translation Model (4 features)	The 4 standard feature: e2f, f2e, lex-e2f, lex-f2e TM-combine 3 models: in-domain, general-domain, and out-domain TM
Language Models (3 features)	in-domain 5-gram LM general-domain 5-gram LM out-domain 5-gram LM
Reordering (5 features)	general domain OSM
Others (4 features)	UnknownWordPenalty, WordPenalty, PhrasePenalty, Top10 standard SparseFeatures
Rescoring (4 features)	subgeneral-domain forward RNNLM in-domain backward RNNLM in-domain S2T bidirectional NNJM in-domain T2S bidirectional NNJM
H2-SMS-CHT (wrt. H1-SMS-CHT)	- Top10 standard SparseFeatures
H1-CTS (wrt. H1-SMS-CHT)	- in-domain 5-gram LM + subgeneral-domain 5-gram LM - in-domain backward RNNLM - in-domain S2T bi-dir. NNJM + general-domain backward RNNLM + subgeneral-domain S2T bi-dir. NNJM
H2-CTS (wrt. H1-CTS)	- Top10 standard SparseFeatures

Table 5: Features for the H1 system on SMS-CHT

2.7.2 Punctuation Correction

We have applied punctuation correction on source language on all the train/tune/test/eval dataset. A Chinese punctuation prediction model is trained using the CRF++ toolkit (Kudo, 2005) on our in-house SMS/Chat text (80%) and OpenMT15 (20%) filtered general-domain data. It is a standard CRF punctuation prediction model based on the work of (Wang et al., 2012). Punctuation correction is performed using the predicted punctuation and the following two heuristics:

1. We enforce the correction mechanism to obey

the following order: *None* → *Comma* → *Period* → *QMark/EMark* (question marks and exclamation marks), i.e., correction can only be done along the direction, but not against the direction.

2. Sentence-end punctuation can only be changed from non-period to question mark.

The above two heuristics are derived from our observations. The first one is because in SMS, people tend to simplify punctuation from *QMark/EMark* to *Period*, to *Comma*, to *None*, probably due to the ease of typing. The second one is because in human translation, if the source does not end with any punctuation, usually the target will not end with any punctuation either, unless it is a question sentence.

2.7.3 Translation Model Combination

We have combined multiple translation models (TM) using minimum-perplexity domain adaptation (Sennrich, 2012). We modified the original script to make it work for hierarchical-phrase-based models. We found that combining TMs trained using different Chinese word segmentation and different domains gives significant improvement. For the alignment, we use GIZA++.

2.7.4 Decoding

We have modified Moses hierarchical decoder (version 2.1.1) to add in the NNJM (Neural Network Joint Model, (Devlin et al., 2014)) feature and the OSM (Operation Sequence Model, (Durrani et al., 2011)) feature. However, our experiments show that for the hier-phrase-based (HPB) system, using both NNJM and OSM in decoding, no further gain is achieved even though using either of them alone, a big gain is achieved. Note that this anomaly does not apply to the phrase-based system. We suspect that the NULL-word estimation using weighted average word vector may introduce some noise, which is small enough so that using NNJM alone we can still get improvement. However, in the presence of OSM, if we want to use NNJM to get further improvement, since the system is already doing very well, it is harder to improve upon it. Thus, it imposes a more stringent requirement on the quality of features. As a result, the noise becomes more significant and thus, no further improvement is achieved in our experiments. Take note that the NULL-word token is only used in HPB system during decoding. Thus, for HPB, we do not use NNJM during decoding, but use them only during N-best rescoring. The decoder is tuned using batch-MIRA algorithm.

2.7.5 N-best Rescoring

For N-best rescoring on complete hypotheses, we have added bilingual-bidirectional NNJM as well as both forward and backward RNNLMs (recurrent neural-network language model) (Mikolov et al., 2010) features. In such a way, the language models will cover all the target words before and after the current word, as well as a span of 11 source words centered at the aligned source word, as shown in Figure 2.

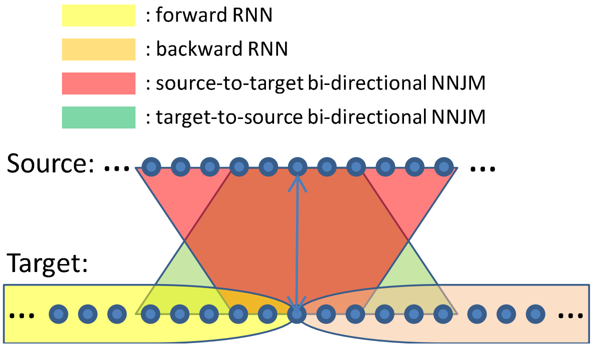
🟨 : forward RNN
🟧 : backward RNN
🟥 : source-to-target bi-directional NNJM
🟩 : target-to-source bi-directional NNJM
Source: …
…
Target:
…
…

Figure 2: Word coverage by neural-network language models for N-best rescoring.

Let s_i denotes i^{th} word relative to current source word, t_i denotes i^{th} word relative to the current target word, for S2T bi-directional NNJM, the neural network computes $P(t_0|t_{-2}, t_{-1}, t_{+1}, t_{+2}, s_{-5}, ..., s_{+5}))$, for T2S bi-directional NNJM, it computes $P(s_0|s_{-2}, t_{-1}, t_{+1}, s_{+2}, t_{-5}, ..., t_{+5}))$.

For NNJM, since our model has a bi-directional word context, we have modified the heuristic for finding alignment points: on the target side, find the closest aligned word (at position tL) to the left of the current word (at position tC), compute its average source alignment position, sL; similarly, find the closest aligned word (at position tR) to the right of the current word (at position tC), compute its average source alignment position, sR; lastly, compute the source alignment point by interpolating the two average positions, sL and sR as shown in Equation 1:

$$sC = sL + (sR - sL) * (tC - tL)/(tR - tL) \quad (1)$$

Experimentally, we found that for bi-directional NNJM, our interpolate alignment heuristic performs slightly better than the right-preferred heuristic in (Durrani et al., 2011).

2.8 System Combination

We use MEMT (Multi-Engine Machine Translation) (Heafield and Lavie, 2010) for system combination. For SMS/Chat, we combine 5 system outputs with 2 from our phrase-based and 3 from our hierarchical phrase-based system. For CTS, we combine 4 systems outputs with 2 from our phrase-based and 2 from our hierarchical-phrase-based system. A 300-best list of combined translations is generated from which we select the top 1 output as the final translation. The language model feature used for both *SMS-CHT* and *CTS* data is a 5-gram language model generated on the sub-general domain data (*SMS-CHT-CTS*) on target side using SRILM. The feature weights are tuned by running Z-MERT. We use a tuning set that is carved out from the non-gold standard tuning data. For SMS/Chat data, the tuning set contains 1505 sentences randomly selected from the *SMS-CHT* tuning data. For CTS data, the tuning set contains 1393 sentences randomly selected from the CTS tuning data.

2.9 Post-processing

The post-processing framework is shown in Figure 3. Only the most significant components are described in this section.

2.9.1 Chinese-to-English Number Translator

Our in-house Chinese-to-English number translator is applied to post-process untranslated Chinese numerals. It is very sophisticated, capable of translating not only Chinese numerals and ordinals, but also Chinese ranged numerals like 五六百(500 to 600), 一百三四十(130 to 140), 十三四(13 to 14), etc., Chinese floating-point numerals like 十点三四(10.34), Chinese time numerals like 十点三十四(10:34), Chinese fractional numerals like 五分之一(one fifth), 三分之二(two thirds), 三点五分之三点四(3.4/3.5) and etc.

2.9.2 Dictionary-based Lexical Translator

Our dictionary-based lexical translator is applied to post-process untranslatable Chinese words. The dictionary is built from Group2-*LEX* (Table 2).

2.9.3 English Recaser

We trained the English recaser using Moses SMT framework with all available conversational data mentioned in Group 1. Default weights are used without any tuning because we found that tuning the recaser will degrade the translation performance instead.

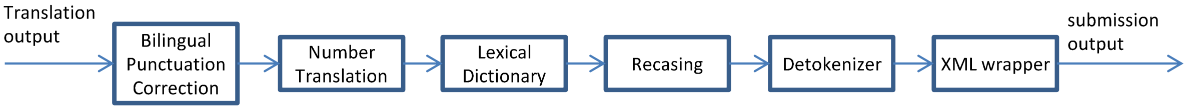

Figure 3: Post-processing pipeline overview.

3 Experiments

3.1 The Effect of Various Features

We have tried various features and techniques. Their effect on the baseline system are shown in Table 6 below. The scores shown are the average scores from 3–8 tuning and testing runs and thus, are quite reliable.

System	BLEU	NIST	METEOR	TER
bl0: baseline	20.48	5.169	26.32	68.10
bl0 + left-NNJM (right)	21.15	5.316	27.09	66.75
bl0 + bi-dir-NNJM (right)	21.30	5.252	27.06	67.14
bl0 + bi-dir-NNJM (interp)	21.50	5.274	27.16	67.15
bl0 + OSM + left-NNJM (right)	21.18	5.283	26.80	67.94
bl0 + sparse-features	20.70	5.309	26.48	66.59
bl0 + punc-correct	20.52	5.197	26.34	67.94
bl0 + punc-correct + split-merge	20.78	5.210	26.50	67.87
bl1: bl0 + gigaword-lm + indomain-lm	20.77	5.295	26.70	67.73
bl1 + OSM	21.41	5.334	26.79	66.46
bl2: bl1 + OSM + TM-combine	22.38	5.454	27.51	65.53
bl2 + sparse	22.81	5.526	27.53	64.95
bl2 + sparse + 4NNLM-rescore	23.69	5.650	28.10	62.42

Table 6: Effects of various features in the hier-phrase-based framework on SMS/Chat. The baseline is trained using general-domain parallel corpus. TM-combine adds a smaller indomain TM and much larger out-domain TM using a different Chinese word segmenter. All NNJMs are source-to-target, left-NNJM means NNJM with left target context, the alignment heuristics is shown in the bracket.

In Table 6, the first 4 data rows show that bi-directional NNJM with interpolate heuristic works slightly better than others. NNJM alone gives a very big improvement on the baseline system. However, it fails to achieve further improvement when OSM is added (5th data row). On the other hand, Punctuation correction with split-and-merge, Gigaword LM and sparse features gives small but consistent improvement. Overall, the biggest improvement still comes from OSM, TM-combine, and rescoring with neural-network LMs.

3.2 System Combination

Our experimental results show that combining similar systems, such as different systems in phrase-based framework, does not give consistent improvement. System combination will benefit machine translation only when complementary systems are combined such as combining systems from both phrase-based and hierarchical phrase-based frameworks. Even when combining complementary systems, there are some exceptions where system combination may not benefit machine translation. If the performance difference between various systems is very large, then it is more difficult for the combined results to surpass the best system.

3.3 Submitted Systems

We have finally submitted the translation output on the test data from 1 primary and 7 contrastive systems as follows:

a) Primary System: the combined system from b), c), d), and e) with bilingual punctuation correction on SMS/Chat

b) Constrastive System 1: the phrase-based decoder+rescoring system, i.e., System P1-SMS-CHT & P1-CTS

c) Constrastive System 2: the phrase-based decoder system with neural network feature, i.e., System P2-SMS-CHT & System P2-CTS

d) Constrastive System 3: the hierarchical-phrase-based decoder+rescoring system with sparse features, i.e., System H1-SMS-CHT and H1-CTS-CHT

e) Constrastive System 4: the hierarchical-phrase-based decoder+rescoring system without sparse features, i.e., System H2-SMS-CHT and H2-CTS-CHT

f) Constrastive System 5: our baseline phrase-based decoder system, it has about the same performance as our baseline hierarchical-phrase-based decoder system.

g) Constrastive System 6: the hierarchical-phrase-based decoder system with sparse features, i.e., System H1-SMS-CHT and H1-CTS-CHT (both without rescoring)

h) Constrastive System 7: the combined system from b), c), d), and e) without bilingual punctuation correction.

System	BLEU	IBMBLEU	NIST	METEOR	TER
primary	18.53	18.40	3.43	26.15	35.51
contrast1	17.53	17.38	3.33	25.67	34.39
contrast2	17.62	17.49	3.34	25.59	33.73
contrast3	17.69	17.57	3.36	25.71	35.15
contrast4	17.49	17.37	3.34	25.63	34.33
contrast5	16.45	16.29	3.30	25.33	33.17
contrast6	16.81	16.69	3.30	25.23	33.29
contrast7	18.54	18.41	3.43	26.15	35.52

Table 7: Official results: averaged scores for SMS, Chat and CTS.

From the results in Table 7, the various techniques we have tried these months (multiple LMs, OSM, NNJM, various reordering models and rescoring) gives about 1-1.3 BLEU score improvement in total including rescoring, [contrast1/2/3/4 vs contrast5]; rescoring alone gives about 0.8 BLEU score improvement, [contrast3 vs contrast6]; system combination gives about 0.6-0.7 BLEU score, [primary vs contrast1/2/3/4]; adding sparse features gives about 0.2 BLEU improvement, [contrast3 vs contrast4].

4 Conclusion

In this paper, we have described the I^{2}R SMT system that was used in the OpenMT15 evaluation campaign. We use a 3-pass approach: decoding, rescoring, and system-combination. The most effective features for decoding and rescoring are in descending order: TM-combination, NNJM, OSM, various reordering models, RNNLM, source punctuation correction with split-and-merge, external language models, top10 sparse features. Moreover, combining systems from different frameworks gives better performance than combining system from the same framework.

References

Jacob Devlin, Rabih Zbib, Zhongqiang Huang, Thomas Lamar, Richard Schwartz, and John Makhoul. 2014. Fast and robust neural network joint models for statistical machine translation. In *52nd Annual Meeting of the Association for Computational Linguistics, Baltimore, MD, USA, June*.

Nadir Durrani, Helmut Schmid, and Alexander Fraser. 2011. A joint sequence translation model with integrated reordering. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 1045–1054. Association for Computational Linguistics.

Chris Dyer, Victor Chahuneau, and Noah A Smith. 2013. A simple, fast, and effective reparameterization of ibm model 2. In *HLT-NAACL*, pages 644–648. Citeseer.

Marcello Federico and Nicola Bertoldi. 2005. A word-to-phrase statistical translation model. *ACM Transactions on Speech and Language Processing (TSLP)*, 2(2):1–24.

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. Association for Computational Linguistics.

Kenneth Heafield and Alon Lavie. 2010. Combining machine translation output with open source: The carnegie mellon multi-engine machine translation scheme. *The Prague Bulletin of Mathematical Linguistics*, 93:27–36.

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. Association for Computational Linguistics.

Philipp Koehn, Amittai Axelrod, Alexandra Birch Mayne, Chris Callison-Burch, Miles Osborne, David Talbot, and Michael White. 2005. Edinburgh system description for the 2005 nist mt evaluation. In *Proceedings of Machine Translation Evaluation Workshop*, volume 2005.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. 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. Association for Computational Linguistics.

Taku Kudo. 2005. CRF++: yet another CRF toolkit. *http://crfpp.sourceforge.net*.

Tomas Mikolov, Martin Karafiát, Lukas Burget, Jan Cernockỳ, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *INTERSPEECH 2010, 11th Annual Conference of the International Speech Communication Association, Makuhari, Chiba, Japan, September 26-30, 2010*, pages 1045–1048.

Franz Josef Och and Hermann Ney. 2000. Giza++: Training of statistical translation models.

Rico Sennrich. 2012. Perplexity minimization for translation model domain adaptation in statistical machine translation. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 539–549. Association for Computational Linguistics.

Xuancong Wang, Hwee Tou Ng, and Khe Chai Sim. 2012. Dynamic conditional random fields for joint sentence boundary and punctuation prediction. In *Proceedings of Interspeech*.