Multi-Source Transformer for Kazakh-Russian-English Neural Machine Translation

Patrick Littell Chi-kiu Lo Samuel Larkin Darlene Stewart
NRC-CNRC
National Research Council of Canada
1200 Montreal Road, Ottawa, Ontario K1A 0R6, Canada
{Patrick.Littell|Chikiu.Lo|Samuel.Larkin|Darlene.Stewart}@nrc-cnrc.gc.ca

Abstract

We describe the neural machine translation (NMT) system developed at the National Research Council of Canada (NRC) for the Kazakh-English news translation task of the Fourth Conference on Machine Translation (WMT19). Our submission is a multi-source NMT system taking both the original Kazakh sentence and its Russian translation as input for translating into English.

1 Introduction

The WMT19 (Bojar et al., 2019) Kazakh-English News Translation task presented a machine translation scenario in which parallel resources between the two languages (˜200k sentences) were considerably fewer than parallel resources between these languages and a third language, Russian (˜14M English-Russian sentence pairs and ˜5M Kazakh-Russian pairs).

The NRC team therefore explored machine translation pipelines that utilized the Russian resources, including:

1. "Pivoting" through Russian: training an MT system from Kazakh to Russian, and another system from Russian to English (Fig. 1a).
2. Creating a synthetic Kazakh-English parallel corpus by training a Russian-Kazakh MT system and using it to "cross-translate"[1] the Russian-English corpus (Fig. 1b).
3. Training a multi-encoder (Libovický and Helcl, 2017; Libovický et al., 2018) Transformer system (Vaswani et al., 2017) from Kazakh/Russian to English that subsumes both of these approaches (Fig. 1c).

Techniques (1) and (2) both involve the translation of genuine data into a synthetic translation (into Russian in the first case, and into Kazakh in the second case). It is, however, possible to attend to *both* the original sentence and its translation using multi-source techniques (Zoph and Knight, 2016; Libovický and Helcl, 2017; Nishimura et al., 2018); we hypothesized that giving the system both the originals and "cross-translations", in both directions (Kazakh-to-Russian and Russian-to-Kazakh), would allow the system to make use of the additional information available by seeing the sources before translation.

Our multi-encoder Transformer approach performed best among our submitted systems by a considerable margin, outperforming pivoting by 4.2 BLEU and augmentation by one-way cross-translation by 10.2 BLEU.[2]

2 Multilingual data

2.1 Kazakh-English

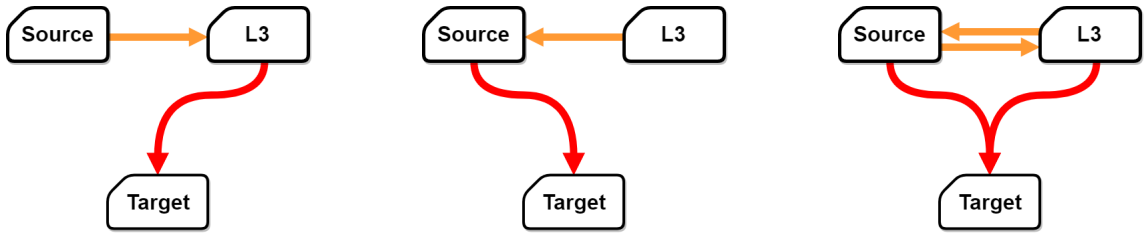
The raw bilingual Kazakh-English data provided for the constrained news translation task consists of web-crawled data, news commentary data and Wikipedia article titles. In total, they account for ˜200k sentence pairs. All these data were used to train the foundation systems for back-translation. Since the web-crawled data is very noisy, we removed all the web-crawled portion from the training data before training our final submitted system.

For tuning and evaluating, we used the `newsdev2019-kken` data set; for SMT, we

[1] We term synthetic data creation by translation between source languages "cross-translation" to distinguish it from "back-translation" in the sense of Sennrich et al. (2016). Nishimura et al. (2018), which also uses source$_1$-to-source$_2$ translation, calls both kinds of synthetic data creation "back-translation", but because our pipeline uses both kinds we distinguish them with separate terms.

[2] However, these systems, as submitted, are not directly comparable due to some additional data filtering in our final submitted system; we will be releasing more direct comparisons and a more thorough description of the architecture in a companion article.

Proceedings of the Fourth Conference on Machine Translation (WMT), Volume 2: Shared Task Papers (Day 1) pages 267–274
Florence, Italy, August 1-2, 2019.



(a) "Pivoting": two systems (source-to-L3 and L3-to-target) executed in a pipeline

(b) Augmentation of source/target corpus with "cross-translated" synthetic data

(c) Multi-source system with augmentation by cross-translation in both directions

Figure 1: Approaches to utilizing a third language ("L3") in machine translation.

split it into two sets as our internal dev and devtest; dev contains 1266 sentence pairs and devtest contains the remaining 800 sentence pairs.

2.2 Kazakh-Russian

The raw bilingual Kazakh-Russian data provided to assist in the news translation task is web-crawled data. In total, they account for ~5M sentence pairs. All these data were used to train the foundation systems for cross-translation.

For tuning and evaluating, we randomly selected 1000 sentence pairs each for the dev and devtest sets from the provided bilingual data. The remaining bilingual data is de-duplicated against the bag of 6-grams collected from the dev and devtest sets. The de-duplicated bilingual data has ~4.2M sentence pairs.

2.3 Russian-English

The raw bilingual Russian-English data we used in our systems consists of web-crawled data, news commentary data and Wikipedia article titles. In total they account for ~14M sentence pairs. All these data were used to train the foundation systems for back-translation. Since the Paracrawl portion of the bilingual data is very noisy, before training our final submitted system we ran our parallel corpus filtering pipeline (Lo et al., 2018) with YiSi-2 as the scoring function (instead of MT + YiSi-1) and trimmed the size of the Paracrawl portion from 12M sentence pairs to 4M sentence pairs.

For tuning and evaluating, we used the `newstest2017-enru` data set as the dev set and the `newstest2018-enru` data set as the devtest set.

3 Data preparation

3.1 Cleaning and tokenization

Our preprocessing pipeline begins by cleaning the UTF-8 with both Moses' cleaning script[3] and an in-house script that performs additional whitespace, hyphen, and control character normalization. We then proceed to normalize and tokenize the sentences with Moses' punctuation normalization[4] and tokenization scripts[5].

3.2 Transliteration

To mitigate some of the overall complexity, and allow greater sharing in joint BPE models and weight tying, we first converted the Kazakh and Russian text from Cyrillic to Roman, using official Romanization standards using spm_normalize (Kudo, 2018) and transliteration tables from Wiktionary for Kazakh[6] and Russian[7].

3.3 Byte-pair encoding

Our BPE model is a joint one across transliterated Kazakh, transliterated Russian, and English. Using fastBPE[8], we created a 90k-operation BPE model, balancing the three languages with ~8.2M sentences of each, using:

- all available Kazakh from bilinugual kk-en;
- all available Kazakh from bilinugual kk-ru;

[3] github.com/moses-smt/mosesdecoder/scripts/tokenizer/remove-non-printing-char.perl

[4] github.com/moses-smt/mosesdecoder/scripts/tokenizer/normalize-punctuation.perl

[5] github.com/moses-smt/mosesdecoder/scripts/tokenizer/tokenizer.perl

[6] en.wiktionary.org/wiki/Module:kk-translit

[7] en.wiktionary.org/wiki/Module:ru-translit

[8] github.com/glample/fastBPE

- all monolingual Kazakh news and wiki data;
- all available English from bilingual kk-en;
- a sample of ˜8M English sentences from bilingual ru-en and monolingual en;
- all available Russian from bilinugual kk-ru;
- a sample of ˜3.2M Russian sentences from bilingual ru-en and monolingual ru.

A separate vocabulary was extracted for each language using the corpora used to create the BPE model. The BPE model was then applied to all training, dev and devtest data.

4 Multi-encoder transformer

We implemented a multi-source Transformer (Vaswani et al., 2017) architecture, in the Sockeye (Hieber et al., 2017) framework, that combines the output of two encoders (one for Kazakh, one for Russian); this architecture will be described in greater detail in a companion paper.

Our encoder combination takes place during attention (that is, the attention step in which information from the decoder and encoders are combined, rather than the self-attention steps inside each encoder and decoder); Figure 2 illustrates the position in which the multiple sources are combined into a single representation.

First, we perform multi-head scaled dot-product attention between the the decoder and each encoder separately.

$$\boldsymbol{C}^{(s)} = \text{MultiHead}^{(s)}\left(\boldsymbol{D}, \boldsymbol{H}^{(s)}, \boldsymbol{H}^{(s)}\right) \quad (1)$$

$$\text{MultiHead}^{(s)}\left(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}\right) = \sum_{i}^{h} \text{Head}_i^{(s)} \boldsymbol{W}_i^{O(s)} \quad (2)$$

$$\text{Head}_i^{(s)}\left(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}, d_k\right) = \mathcal{A}(\boldsymbol{Q}\boldsymbol{W}_i^{Q(s)}, \boldsymbol{K}\boldsymbol{W}_i^{K(s)}, \boldsymbol{V}\boldsymbol{W}_i^{V(s)}, d_k) \quad (3)$$

$$\mathcal{A}\left(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}, d_k\right) = \text{softmax}\left(\frac{\boldsymbol{Q}\boldsymbol{K}^\top}{\sqrt{d_k}}\right)\boldsymbol{V} \quad (4)$$

where $\boldsymbol{D} = (\boldsymbol{d}_1, \boldsymbol{d}_2, \cdots, \boldsymbol{d}_n)$, $\boldsymbol{d}_i \in \mathbb{R}^{d_{model}}$ represents the decoder states, $\boldsymbol{H} = (\boldsymbol{h_1}, \boldsymbol{h_2}, \cdots, \boldsymbol{h_m})$, $\boldsymbol{h_i} \in \mathbb{R}^{d_{model}}$ represents the outputs of the encoder's final self-attention layer, $\boldsymbol{W}_i^{Q(s)} \in \mathbb{R}^{d_{model} \times d_k}$, $\boldsymbol{W}_i^{K(s)} \in \mathbb{R}^{d_{model} \times d_k}$, $\boldsymbol{W}_i^{V(s)} \in \mathbb{R}^{d_{model} \times d_k}$ and $\boldsymbol{W}_i^{O(s)} \in \mathbb{R}^{d_k \times d_{model}}$ are trainable parameter matrices which project the key, query and value into a smaller dimensionality. Together with $d_k = d_{model}/h$, we have $\boldsymbol{C}^{(s)} \in \mathbb{R}^{n \times d_{model}}$.

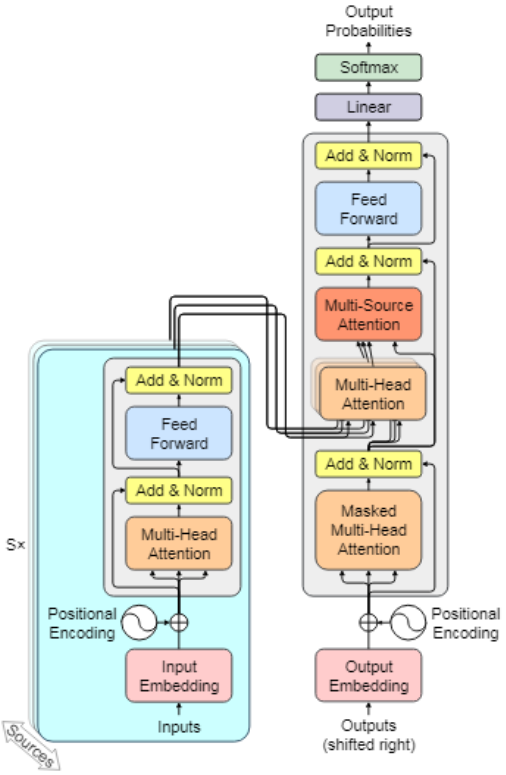


Figure 2: Multi-source attention on $\boldsymbol{S}$ sources. Each output from the $\boldsymbol{S}$ encoders is attended to by a separate multi-head attention layer (Eqs. 1-4), and then the outputs of these attention layers are combined (Eq. 5).

Next, we combine the outputs from the different encoders with a simple projection and sum, similar to what Libovický et al. (2018) refer to as "parallel":

$$\tilde{C} = \sum_{i}^{S} \boldsymbol{C}^{(i)} \boldsymbol{W}^{C(i)} \quad (5)$$

As this is essentially the same operation as the multi-head combination in Equation (2), and no nonlinearities intervene, we can also conceptualize Equations (1)-(5) as if they were a single multi-head attention layer with $S * h$ heads (in this case $2 * 8$ heads), in which each group of h heads is constrained to attend to the output of one encoder.

We also experimented with a hierarchical attention mechanism along the lines of Libovický and Helcl (2017) and Libovický et al. (2018), but as this did not outperform the simpler combination mechanism in (5) in internal testing, our submitted systems utilized the latter.

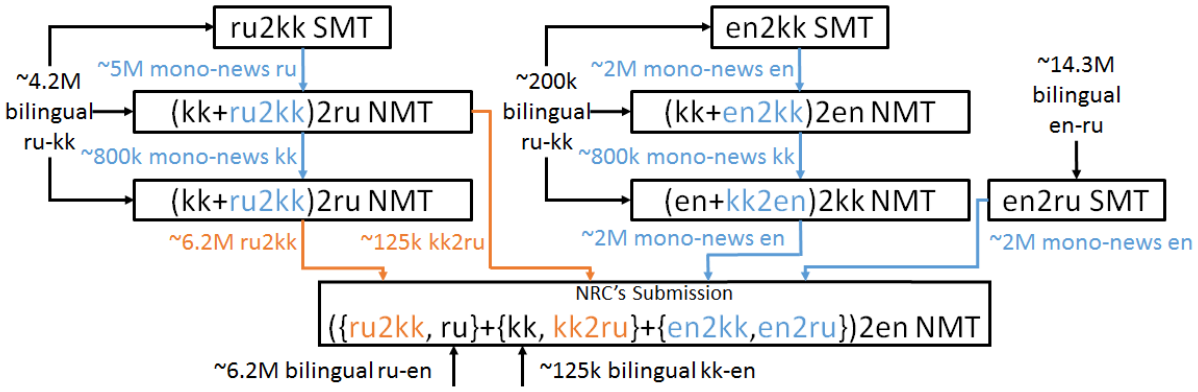


Figure 3: The relations of all the MT systems involved in building the NRC final submitted system.

5 Experiments and results

5.1 NMT Setup

Our code extends sockeye-1.18.72 from Hieber et al. (2017). Each source encoder has 6 layers and our decoder also has 6 layers, with a model dimension of $d_{model} = 512$ and 2048 hidden units sub-layer feed-forward networks. We use weight tying, where the source embeddings, the target embeddings and the target softmax weights are tied, which implies a shared vocab. We trained employing a cross-entropy loss with Adam (Kingma and Ba, 2014), $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 1e-8$ and an initial learning rate of 0.0001, decreasing the learning by 0.7 each time the development-set BLEU did not improve for 8 checkpoints. We optimized against BLEU using `newsdev2019-kken` as the development set, stopping early if BLEU did not improve for 32 checkpoints of 1000 updates each. The inputs and output lengths were restricted to a maximum of 60 tokens, and mini-batches were of variable size depending on sentence length, with each mini-batch containing up to 4096 words.

5.2 SMT Setup

We trained en2kk, ru2kk and en2ru SMT systems using Portage (Larkin et al., 2010), a conventional log-linear phrase-based SMT system, using the corresponding BPEed parallel corpora prepared as described in Section 3. The translation model of each SMT system uses IBM4 word alignments (Brown et al., 1993) with grow-diag-final-and phrase extraction heuristics (Koehn et al., 2003). The systems each have two n-gram language models: a 5-gram language model (LM) (a mixture LM in the kk2en case) trained on the target-side of the corresponding parallel corpora using SRILM (Stolcke, 2002), and a pruned 6-gram LM trained on the monolingual training corpora (for en2ru, trained just on news using KenLM (Heafield, 2011); for ru2kk and en2kk, a static mixture LM trained on all monolingual Kazakh data using SRILM). Each SMT system also includes a hierachical distortion model, a sparse feature model consisting of the standard sparse features proposed in Hopkins and May (2011) and sparse hierarchical distortion model features proposed in Cherry (2013), and a neural network joint model, or NNJM, with 3 words of target context and 11 words of source context, effectively a 15-gram LM (Vaswani et al., 2013; Devlin et al., 2014). The parameters of the log-linear model were tuned by optimizing BLEU on the development set using the batch variant of the margin infused relaxed algorithm (MIRA) by Cherry and Foster (2012). Decoding uses the cube-pruning algorithm of Huang and Chiang (2007) with a 7-word distortion limit.

We then used these SMT systems to back-translate a ˜2M sentence subselection of monolingual English news into Kazakh and Russian, and a ˜5M sentence subselection of monolingual Russian news into Kazakh, as well as cross-translating the Russian of the ru-en parallel corpora into Kazakh.

5.3 Building the NRC Submission System

Our final submission involved several SMT components and several NMT components to produce back-translations and cross-translations needed for our multi-source submission system, as shown in Figure 3.

Available Resources	Training Source 1	Training Source 2	Training Att. Comb.	Dev./Test Source 1	Dev./Test Source 2	BLEU Dev.	BLEU Test
kk-en	kk+en2kk	–	–	kk	–	12.8	9.9
kk-en, ru-en	kk+ru+en2kk	–	–	kk	–	15.4	12.6
kk-en, kk-ru, ru-en	kk+ru2kk+en2kk	–	–	kk	–	17.9	14.8
kk-ru, ru-en	pivoting					19.3	20.8
kk-en, kk-ru, ru-en	kk+ru2kk+en2kk	kk2ru+ru+en2ru	Parallel	kk	kk2ru	19.6	**24.2 /25.0***

Table 1: BLEU scores on WMT19 Kazakh-English news translation. en2kk denotes synthetic Kazakh back-translated from English. ru2kk denotes synthetic Kazakh cross-translated from Russian. en2ru denotes synthetic Russian back-translated from English. kk2ru denotes synthetic Russian cross-translated from Kazakh. * denotes an unofficial post-competition result, a fully-trained version of our top system, which had only been partially trained due to time constraints.

5.3.1 Synthetic cross-translations

To synthesize cross-translations, we trained 3 systems using our filtered ˜4.2M sentences of bilingual Russian-Kazakh data. First, we trained a Russian-to-Kazakh (ru2kk) SMT system and then used it to generate ˜5M sentences of synthetic Kazakh. Augmenting the bilingual data with the Kazakh back-translations, we trained a Kazakh-to-Russian NMT system to back translate ˜800k sentences of monolingual Kazakh news for a ru2kk NMT system and to cross translate ˜125k kk-en sentences for one component of our final system. Finally, we trained a Russian-to-Kazkah NMT system using the bilingual data and the synthetic Russian to cross translate ˜6M for our second component of the final system.

5.3.2 Synthetic back-translation

A stack of another three MT systems was used to synthesize Kazakh from English using ˜200k of available English-Kazakh bilingual data for training. Starting with an English-to-Kazakh SMT system, ˜2M English sentences were back-translated to Kazakh. Augmenting the bilingual data with the newly generated Kazakh, we trained a NMT Kazakh-to-English system and back translated ˜800k sentences of Kazakh news. The last English-to-Kazakh NMT system in that stack was trained using the bilingual data enlarged with the ˜800k previously generated back-translations. It generated our en2kk back-translation of ˜2M sentences of English news.

Our final component was accomplished by training an English-to-Russian SMT system using ˜14.3M bilingual sentences and back translating the ˜2M sentence subselection of English news into Russian.

5.3.3 Putting it all together

The box labelled "NRC's Submission" in Figure 3 depicts how each sub-corpus was assembled into the final bilingual corpora used to train our multi-source NMT submission system. Each set of curly braces surrounds a pair of corresponding Kazakh and Russian sources. The first pair represents Kazakh and its cross-translation to Russian, the second is the cross-translation of Russian-to-Kazakh with the original Russian, and lastly we have our sub-selected corpus back-translated into both Kazakh and Russian.

5.4 Results

We can see in Table 1 that the full multi-source, multi-encoder system with two-way cross-translation (both Kazakh-to-Russian and Russian-to-Kazakh) is significantly better than our other systems, outperforming the pivoting system (on the fourth line) by 4.2 BLEU and augmentation by one-way cross-translation (on the third line) by 10.2 BLEU.

We believe this improvement over the other two methods is due to the model being able to attend to additional original data, to which the other systems do not have direct access. Both pivoting and one-way synthetic augmentation involve "discarding" genuine data, in that some of the original sentences – Kazakh sentences in the former, and Russian sentences in the later – are never seen by the downstream system, since they are only encountered in translation. Multi-source methods allow a system to attend to the original data in both directions, thus capturing information that would otherwise be lost in translation.

Notable in this table is the comparative improvement of the test scores over the dev scores, between the pivoting (line 4) and multi-source (line 5) systems. This can be explained, we

System	BLEU	YiSi-1	YiSi-1_srl
NEU	30.5	79.19	76.97
rug-morfessor	27.9	77.70	75.47
talp-upc-2019	24.9	75.07	72.74
NRC-CNRC	**24.9**	**75.76**	**73.41**
Frank-s-MT	19.8	76.17	73.87

Table 2: Automatic evaluation results for the top 5 constrained systems in WMT19

System	Ave	Ave. B
NEU	70.1	0.218
rug-morfessor	69.7	0.189
talp-upc-2019	67.1	0.113
NRC-CNRC	**67.0**	**0.092**
Frank-s-MT	65.8	0.066

Table 3: Human evaluation results for the top 5 constrained systems in WMT19

think, by a domain difference between the dev and test sets, where the dev set was sampled from the same news commentary dataset as the training data, whereas the test set comes from actual newswire text. The scores appear to show that the multi-source system has managed to generalize better to newswire text, possibly because it has seen synthetic newswire text (synthesized from the English-Russian dataset) and can respond more appropriately to it.[9]

Tables 2 and 3 compare our multi-source system to the other official submissions in the top 5 of the WMT19 competition. In automatic evaluation by BLEU, we were tied for third place, although with a slight edge when measured by YiSi-1 (Lo, 2019); in human evaluation, we were in a statistical tie for second place. Notably, our multi-source system was the top non-ensemble pure NMT system, with other higher-scoring systems either being ensembles or SMT/NMT hybrids.

6 Conclusion and future work

We present the NRC submission to the WMT19 Kazakh-English news translation shared task. Our submitted system is a multi-source, multi-encoder neural machine translation system that takes Russian as the second source in the system. The advantages of using the multi-source NMT architecture are that it incorporates additional information obtained from 1) the Russian-English training data cross translated into Kazakh, and 2) the Russian cross translated from Kazakh in the Kazakh-Russian training data.

The drawback of this approach is the comparative complexity of the pipeline, with separate systems being trained to create back-translations and cross-translations (including back-translations to train those systems themselves). This complexity was difficult for a human team to manage when considered for three languages; it would be prohibitive (without additional automation) when making systems that involve four or more languages. Making use of the multi-source architecture itself for creating back- and cross-translations together, and sharing encoders and decoders between systems that share languages, would considerably lessen the the complexity of the pipeline and the number of distinct systems that need to be trained.

In other future work, we want to consider additional methods of multi-source attention, as well as other means of creating cross-linguistic synthetic data beyond machine translation, for lower-resource language pairs that do not have substantial parallel data but may be, for example, closely related.

[9]Note that, although we did perform additional filtering on the training data of the multi-source system, we do not believe this is the cause of the better performance on the test compared to the pivoting system. In later tests, we found the pivoting system to be relatively insensitive to this filtering process, giving similar BLEU on both dev and test.

References

Ondřej Bojar, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, Christof Monz, Mathias Müller, and Matt Post. 2019. Findings of the 2019 conference on machine translation (wmt19). In *Proceedings of the Fourth Conference on Machine Translation, Volume 2: Shared Task Papers*, Florence, Italy. Association for Computational Linguistics.

Peter F. Brown, Stephen 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.

Colin Cherry. 2013. Improved reordering for phrase-based translation using sparse features. In *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, June 9-14, 2013, Westin Peachtree Plaza Hotel, Atlanta, Georgia, USA*, pages 22–31. The Association for Computational Linguistics.

Colin Cherry and George F. Foster. 2012. Batch tuning strategies for statistical machine translation. In

Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, June 3-8, 2012, Montréal, Canada, pages 427–436. The Association for Computational Linguistics.

Jacob Devlin, Rabih Zbib, Zhongqiang Huang, Thomas Lamar, Richard M. Schwartz, and John Makhoul. 2014. Fast and robust neural network joint models for statistical machine translation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22-27, 2014, Baltimore, MD, USA, Volume 1: Long Papers*, pages 1370–1380. The Association for Computer Linguistics.

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.

Felix Hieber, Tobias Domhan, Michael Denkowski, David Vilar, Artem Sokolov, Ann Clifton, and Matt Post. 2017. Sockeye: A toolkit for neural machine translation. *CoRR*, abs/1712.05690.

Mark Hopkins and Jonathan May. 2011. Tuning as ranking. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, EMNLP 2011, 27-31 July 2011, John McIntyre Conference Centre, Edinburgh, UK, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1352–1362. ACL.

Liang Huang and David Chiang. 2007. Forest rescoring: Faster decoding with integrated language models. In *ACL 2007, Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics, June 23-30, 2007, Prague, Czech Republic*. The Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. Cite arxiv:1412.6980Comment: Published as a conference paper at the 3rd International Conference for Learning Representations, San Diego, 2015.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics, HLT-NAACL 2003, Edmonton, Canada, May 27 - June 1, 2003*. The Association for Computational Linguistics.

Taku Kudo. 2018. Subword regularization: Improving neural network translation models with multiple subword candidates. *CoRR*, abs/1804.10959.

Samuel Larkin, Boxing Chen, George Foster, Ulrich Germann, Eric Joanis, Howard Johnson, and Roland Kuhn. 2010. Lessons from nrc's portage system at wmt 2010. In *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*, WMT '10, pages 127–132, Stroudsburg, PA, USA. Association for Computational Linguistics.

Jindřich Libovický and Jindřich Helcl. 2017. Attention strategies for multi-source sequence-to-sequence learning. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 196–202, Vancouver, Canada. Association for Computational Linguistics.

Jindřich Libovický, Jindřich Helcl, and David Mareček. 2018. Input combination strategies for multi-source transformer decoder. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 253–260, Belgium, Brussels. Association for Computational Linguistics.

Chi-kiu Lo. 2019. YiSi - A unified semantic MT quality evaluation and estimation metric for languages with different levels of available resources. In *Proceedings of the Fourth Conference on Machine Translation, Volume 2: Shared Task Papers*, Florence, Italy. Association for Computational Linguistics.

Chi-kiu Lo, Michel Simard, Darlene Stewart, Samuel Larkin, Cyril Goutte, and Patrick Littell. 2018. Accurate semantic textual similarity for cleaning noisy parallel corpora using semantic machine translation evaluation metric: The NRC supervised submissions to the parallel corpus filtering task. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 908–916, Belgium, Brussels. Association for Computational Linguistics.

Yuta Nishimura, Katsuhito Sudoh, Graham Neubig, and Satoshi Nakamura. 2018. Multi-source neural machine translation with data augmentation. *CoRR*, abs/1810.06826.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.

Andreas Stolcke. 2002. SRILM - an extensible language modeling toolkit. In *7th International Conference on Spoken Language Processing, ICSLP2002 - INTERSPEECH 2002, Denver, Colorado, USA, September 16-20, 2002*. ISCA.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Ashish Vaswani, Yinggong Zhao, Victoria Fossum, and David Chiang. 2013. Decoding with large-scale neural language models improves translation. In *Proceedings of the 2013 Conference on Empirical*

Methods in Natural Language Processing, EMNLP 2013, 18-21 October 2013, Grand Hyatt Seattle, Seattle, Washington, USA, A meeting of SIGDAT, a Special Interest Group of the ACL, pages 1387–1392. ACL.

Barret Zoph and Kevin Knight. 2016. Multi-source neural translation. In *Proceedings of NAACL-HLT*, pages 30–34.