Syntactically Guided Neural Machine Translation

Felix Stahlberg[†] and **Eva Hasler**[†] and **Aurelien Waite**[‡] and **Bill Byrne**[‡†]

[†]Department of Engineering, University of Cambridge, UK

[‡]SDL Research, Cambridge, UK

Abstract

We investigate the use of hierarchical phrase-based SMT lattices in end-to-end neural machine translation (NMT). Weight pushing transforms the Hiero scores for complete translation hypotheses, with the full translation grammar score and full n-gram language model score, into posteriors compatible with NMT predictive probabilities. With a slightly modified NMT beam-search decoder we find gains over both Hiero and NMT decoding alone, with practical advantages in extending NMT to very large input and output vocabularies.

1 Introduction

We report on investigations motivated by the idea that the structured search spaces defined by syntactic machine translation approaches such as Hiero (Chiang, 2007) can be used to guide Neural Machine Translation (NMT) (Kalchbrenner and Blunsom, 2013; Sutskever et al., 2014; Cho et al., 2014; Bahdanau et al., 2015). NMT and Hiero have complementary strengths and weaknesses and differ markedly in how they define probability distributions over translations and what search procedures they use.

The NMT encoder-decoder formalism provides a probability distribution over translations $\mathbf{y} = y_1^T$ of a source sentence $\mathbf{x}$ as (Bahdanau et al., 2015)

$$P(y_1^T|\mathbf{x}) = \prod_{t=1}^{T} P(y_t|y_1^{t-1}, \mathbf{x}) = \prod_{t=1}^{T} g(y_{t-1}, s_t, c_t) \tag{1}$$

where $s_t = f(s_{t-1}, y_{t-1}, c_t)$ is a decoder state variable and c_t is a context vector depending on the source sentence and the attention mechanism.

This posterior distribution is potentially very powerful, however it does not easily lend itself to sophisticated search procedures. Decoding is done by 'beam search to find a translation that approximately maximizes the conditional probability' (Bahdanau et al., 2015). Search looks only one word ahead and no deeper than the beam.

Hiero defines a synchronous context-free grammar (SCFG) with rules: $X \rightarrow \langle\alpha, \gamma\rangle$, where α and γ are strings of terminals and non-terminals in the source and target languages. A target language sentence $\mathbf{y}$ can be a translation of a source language sentence $\mathbf{x}$ if there is a derivation D in the grammar which yields both $\mathbf{y}$ and $\mathbf{x}$: $\mathbf{y} = \mathbf{y}(D)$, $\mathbf{x} = \mathbf{x}(D)$. This defines a regular language $\mathcal{Y}$ over strings in the target language via a projection of the sentence to be translated: $\mathcal{Y} = \{\mathbf{y}(D) : \mathbf{x}(D) = \mathbf{x}\}$ (Iglesias et al., 2011; Allauzen et al., 2014). Scores are defined over derivations via a log-linear model with features $\{\phi_i\}$ and weights λ. The decoder searches for the translation $\mathbf{y}(D)$ in $\mathcal{Y}$ with the highest derivation score $S(D)$ (Chiang, 2007, Eq. 24) :

$$\hat{\mathbf{y}} = \mathbf{y}\left(\underset{D:\mathbf{x}(D)=\mathbf{x}}{\operatorname{argmax}} \underbrace{P_G(D)P_{LM}(\mathbf{y}(D))^{\lambda_{LM}}}_{S(D)}\right) \tag{2}$$

where P_{LM} is an n-gram language model and $P_G(D) \propto \prod_{(X\rightarrow\langle\gamma,\alpha\rangle)\in D} \prod_i \phi_i(X \rightarrow \langle\gamma, \alpha\rangle)^{\lambda_i}$.

Hiero decoders attempt to avoid search errors when combining the translation and language model for the translation hypotheses (Chiang, 2007; Iglesias et al., 2009). These procedures search over a vast space of translations, much larger than is considered by the NMT beam search. However the Hiero context-free grammars that make efficient search possible are weak models of translation. The basic Hiero formalism can be extended through 'soft syntactic constraints' (Venugopal et al., 2009; Marton and Resnik, 2008) or by

Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, pages 299–305,
Berlin, Germany, August 7-12, 2016. ©2016 Association for Computational Linguistics

adding very high dimensional features (Chiang et al., 2009), however the translation score assigned by the grammar is still only the product of probabilities of individual rules. From the modelling perspective, this is an overly strong conditional independence assumption. NMT clearly has the potential advantage in incorporating long-term context into translation scores.

NMT and Hiero differ in how they 'consume' source words. Hiero applies the translation rules to the source sentence via the CYK algorithm, with each derivation yielding a complete and unambiguous translation of the source words. The NMT beam decoder does not have an explicit mechanism for tracking source coverage, and there is evidence that may lead to both 'over-translation' and 'under-translation' (Tu et al., 2016).

NMT and Hiero also differ in their internal representations. The NMT continuous representation captures morphological, syntactic and semantic similarity (Collobert and Weston, 2008) across words and phrases. However, extending these representations to the large vocabularies needed for open-domain MT is an open area of research (Jean et al., 2015a; Luong et al., 2015; Sennrich et al., 2015; Chitnis and DeNero, 2015). By contrast, Hiero (and other symbolic systems) can easily use translation grammars and language models with very large vocabularies (Heafield et al., 2013; Lin and Dyer, 2010). Moreover, words and phrases can be easily added to a fully-trained symbolic MT system. This is an important consideration for commercial MT, as customers often wish to customise and personalise SMT systems for their own application domain. Adding new words and phrases to an NMT system is not as straightforward, and it is not clear that the advantages of the continuous representation can be extended to the new additions to the vocabularies.

NMT has the advantage of including long-range context in modelling individual translation hypotheses. Hiero considers a much bigger search space, and can incorporate n-gram language models, but a much weaker translation model. In this paper we try to exploit the strengths of each approach. We propose to guide NMT decoding using Hiero. We show that restricting the search space of the NMT decoder to a subset of $\mathcal{Y}$ spanned by Hiero effectively counteracts NMT modelling errors. This can be implemented by generating translation lattices with Hiero, which are then rescored by the NMT decoder. Our approach addresses the limited vocabulary issue in NMT as we replace NMT OOVs with lattice words from the much larger Hiero vocabulary. We also find good gains from neural and Kneser-Ney n-gram language models.

2 Syntactically Guided NMT (SGNMT)

2.1 Hiero Predictive Posteriors

The Hiero decoder generates translation hypotheses as weighted finite state acceptors (WFSAs), or lattices, with weights in the tropical semiring. For a translation hypothesis $\mathbf{y}(D)$ arising from the Hiero derivation D, the path weight in the WFSA is $-\log S(D)$, after Eq. 2. While this representation is correct with respect to the Hiero translation grammar and language model scores, having Hiero scores at the path level is not convenient for working with the NMT system. What we need are predictive probabilities in the form of Eq. 1.

The Hiero WFSAs are determinised and minimised with epsilon removal under the tropical semiring, and weights are pushed towards the initial state under the log semiring (Mohri and Riley, 2001). The resulting transducer is stochastic in the log semiring, i.e. the log sum of the arc log probabilities leaving a state is 0 $(=\log 1)$. In addition, because the WFSA is deterministic, there is a unique path leading to every state, which corresponds to a unique Hiero translation prefix. Suppose a path to a state accepts the translation prefix y_1^{t-1}. An outgoing arc from that state with symbol y has a weight that corresponds to the (negative log of the) conditional probability

$$P_{Hiero}(y_t = y|y_1^{t-1}, \mathbf{x}). \tag{3}$$

This conditional probability is such that for a Hiero translation $y_1^T = \mathbf{y}(D)$ accepted by the WFSA

$$P_{Hiero}(y_1^T) = \prod_{t=1}^{T} P_{Hiero}(y_t|y_1^{t-1}, \mathbf{x}) \propto S(D). \tag{4}$$

The Hiero WFSAs have been transformed so that their arc weights have the negative log of the conditional probabilities defined in Eq. 3. All the probability mass of this distribution is concentrated on the Hiero translation hypotheses. The complete translation and language model scores computed over the entire Hiero translations are pushed as far forward in the WFSAs as possible. This is commonly done for left-to-right decoding in speech recognition (Mohri et al., 2002).

2.2 NMT–Hiero Decoding

As above, suppose a path to a state in the WFSA accepts a Hiero translation prefix y_1^{t-1}, and let y_t be a symbol on an outgoing arc from that state. We define the joint NMT+Hiero score as

$$\log P(y_t|y_1^{t-1},\mathbf{x}) = \\ \lambda_{Hiero} \log P_{Hiero}(y_t|y_1^{t-1},\mathbf{x}) + \\ \lambda_{NMT} \begin{cases} \log P_{NMT}(y_t|y_1^{t-1},\mathbf{x}) & y_t \in \Sigma_{NMT} \\ \log P_{NMT}(\texttt{unk}|y_1^{t-1},\mathbf{x}) & y_t \notin \Sigma_{NMT} \end{cases} \tag{5}$$

Note that the NMT-HIERO decoder only considers hypotheses in the Hiero lattice. As discussed earlier, the Hiero vocabulary can be much larger than the NMT output vocabulary Σ_{NMT}. If a Hiero translation contains a word not in the NMT vocabulary, the NMT model provides a score and updates its decoder state as for an unknown word.

Our decoding algorithm is a natural extension of beam search decoding for NMT. Due to the form of Eq. 5 we can build up hypotheses from left-to-right on the target side. Thus, we can represent a partial hypothesis $h = (y_1^t, h_s)$ by a translation prefix y_1^t and an accumulated score h_s. At each iteration we extend the current hypotheses by one target token, until the best scoring hypothesis reaches a final state of the Hiero lattice. We refer to this step as *node expansion*, and in Sec. 3.1 we report the number of node expansions per sentence, as an indication of computational cost.

We can think of the decoding algorithm as breath-first search through the translation lattices with a limited number of active hypotheses (a beam). Rescoring is done on-the-fly: as the decoder traverses an edge in the WFSA, we update its weight by Eq. 5. The output-synchronous characteristic of beam search enables us to compute the NMT posteriors only once for each history based on previous calculations.

Alternatively, we can think of the algorithm as NMT decoding with revised posterior probabilities: instead of selecting the most likely symbol y_t according the NMT model, we adjust the NMT posterior with the Hiero posterior scores and delete NMT entries that are not allowed by the lattice. This may result in NMT choosing a different symbol, which is then fed back to the neural network for the next decoding step.

	Train set		Dev set		Test set	
	en	de	en	de	en	de
# sentences	4.2M		6k		2.7k	
# word tokens	106M	102M	138k	138k	62k	59k
# unique words	647k	1.5M	13k	20k	9k	13k
OOV (Hiero)	0.0%	0.0%	0.8%	1.6%	1.0%	2.0%
OOV (NMT)	1.6%	5.5%	2.5%	7.5%	3.1%	8.8%
	en	fr	en	fr	en	fr
# sentences	12.1M		6k		3k	
# word tokens	305M	348M	138k	155k	71k	81k
# unique words	1.6M	1.7M	14k	17k	10k	11k
OOV (Hiero)	0.0%	0.0%	0.6%	0.6%	0.4%	0.4%
OOV (NMT)	3.5%	3.8%	4.5%	5.3%	5.0%	5.3%

Table 1: Parallel texts and vocabulary coverage on *news-test2014*.

3 Experimental Evaluation

We evaluate SGNMT on the WMT *news-test2014* test sets (the filtered version) for English-German (En-De) and English-French (En-Fr). We also report results on WMT *news-test2015* En-De.

The En-De training set includes *Europarl v7*, *Common Crawl*, and *News Commentary v10*. Sentence pairs with sentences longer than 80 words or length ratios exceeding 2.4:1 were deleted, as were *Common Crawl* sentences from other languages (Shuyo, 2010). The En-Fr NMT system was trained on preprocessed data (Schwenk, 2014) used by previous work (Sutskever et al., 2014; Bahdanau et al., 2015; Jean et al., 2015a), but with truecasing like our Hiero baseline. Following (Jean et al., 2015a), we use *news-test2012* and *news-test2013* as a development set. The NMT vocabulary size is 50k for En-De and 30k for En-Fr, taken as the most frequent words in training (Jean et al., 2015a). Tab. 1 provides statistics and shows the severity of the OOV problem for NMT.

The BASIC NMT system is built using the Blocks framework (van Merriënboer et al., 2015) based on the Theano library (Bastien et al., 2012) with standard hyper-parameters (Bahdanau et al., 2015): the encoder and decoder networks consist of 1000 gated recurrent units (Cho et al., 2014). The decoder uses a single maxout (Goodfellow et al., 2013) output layer with the feed-forward attention model (Bahdanau et al., 2015).

The En-De Hiero system uses rules which encourage verb movement (de Gispert et al., 2010). The rules for En-Fr were extracted from the full data set available at the WMT'15 website using a shallow-1 grammar (de Gispert et al., 2010). 5-gram Kneser-Ney language models (KN-LM) for the Hiero systems were trained on WMT'15 parallel and monolingual data (Heafield et al., 2013).

(Jean et al., 2015a, Tab. 2)		SGNMT	
Setup	**BLEU**	**Setup**	**BLEU**
BASIC NMT	16.46	BASIC NMT	16.31
NMT-LV	16.95	HIERO	19.44
+ UNK Replace	18.89	NMT-HIERO	20.69
–	–	+ Tuning	21.43
+ Reshuffle	19.40	+ Reshuffle	21.87
+ Ensemble	21.59		

(a) English-German

(Jean et al., 2015a, Tab. 2)		SGNMT	
Setup	**BLEU**	**Setup**	**BLEU**
BASIC NMT	29.97	BASIC NMT	30.42
NMT-LV	33.36	HIERO	32.86
+ UNK Replace	34.11	NMT-HIERO	35.37
–	–	+ Tuning	36.29
+ Reshuffle	34.60	+ Reshuffle	36.61
+ Ensemble	37.19		

(b) English-French

Table 2: BLEU scores on *news-test2014* calculated with `multi-bleu.perl`. NMT-LV refers to the RNNSEARCH-LV model from (Jean et al., 2015a) for large output vocabularies.

	Search space	Vocab.	NMT scores	Grammar scores	KN-LM scores	NPLM scores	# of node expansions per sen.	BLEU (single)	BLEU (ensemble)
1	Lattice	Hiero		✓	✓		–	21.1 (Hiero)	
2	Lattice	Hiero		✓	✓	✓	–	21.7 (Hiero)	
3	Unrestricted	NMT	✓				254.8	19.5	21.8
4	100-best	Hiero	✓				2,233.6 (DFS: 832.1)	22.8	23.3
5	100-best	Hiero	✓	✓	✓			22.9	23.4
6	100-best	Hiero	✓	✓	✓	✓		22.9	23.3
7	1000-best	Hiero	✓				21,686.2 (DFS: 6,221.8)	23.3	23.8
8	1000-best	Hiero	✓	✓	✓			23.4	23.9
9	1000-best	Hiero	✓	✓	✓	✓		23.5	24.0
10	Lattice	NMT	✓				243.3	20.3	21.4
11	Lattice	Hiero	✓				243.3	23.0	24.2
12	Lattice	Hiero	✓	✓			243.3	23.0	24.2
13	Lattice	Hiero	✓		✓		240.5	23.4	24.5
14	Lattice	Hiero	✓	✓	✓		243.9	23.4	24.4
15	Lattice	Hiero	✓	✓	✓	✓	244.3	24.0	24.4
16	Neural MT – UMontreal-MILA (Jean et al., 2015b)							22.8	25.2

Table 3: BLEU English-German *news-test2015* scores calculated with `mteval-v13a.pl`.

Our SGNMT system[1] is built with the Pyfst interface [2] to OpenFst (Allauzen et al., 2007).

3.1 SGNMT Performance

Tab. 2 compares our combined NMT+Hiero decoding with NMT results in the literature. We use a beam size of 12. In En-De and in En-Fr, we find that our BASIC NMT system performs similarly (within 0.5 BLEU) to previously published results (16.31 vs. 16.46 and 30.42 vs. 29.97).

In NMT-HIERO, decoding is as described in Sec. 2.2, but with $\lambda_{Hiero} = 0$. The decoder searches through the Hiero lattice, ignoring the Hiero scores, but using Hiero word hypotheses in place of any UNKs that might have been produced by NMT. The results show that NMT-HIERO is much more effective in fixing NMT OOVs than the 'UNK Replace' technique (Luong et al., 2015); this holds in both En-De and En-Fr.

For the NMT-HIERO+TUNING systems, lattice MERT (Macherey et al., 2008) is used to optimise λ_{Hiero} and λ_{NMT} on the tuning sets. This yields further gains in both En-Fr and En-De, suggesting that in addition to fixing UNKs, the Hiero predictive posteriors can be used to improve the NMT translation model scores.

Tab. 3 reports results of our En-De system with reshuffling and tuning on *news-test2015*. BLEU scores are directly comparable to WMT'15 results [3]. By comparing row 3 to row 10, we see that constraining NMT to the search space defined by the Hiero lattices yields an improvement of +0.8 BLEU for single NMT. If we allow Hiero to fix NMT UNKs, we see a further +2.7 BLEU gain (row 11). The majority of gains come from fixing UNKs, but there is still improvement from the constrained search space for single NMT.

We next investigate the contribution of the Hiero system scores. We see that, once lattices are generated, the KN-LM contributes more to rescoring than the Hiero grammar scores (rows 12-14). Further gains can be achieved by adding a feed-forward neural language model with NPLM (Vaswani et al., 2013) (row 15). We observe that n-best list rescoring with NMT (Neubig et al., 2015) also outperforms both the Hiero and NMT

[1] http://ucam-smt.github.io/sgnmt/html/

[2] https://pyfst.github.io/

[3] http://matrix.statmt.org/matrix/systems_list/1774

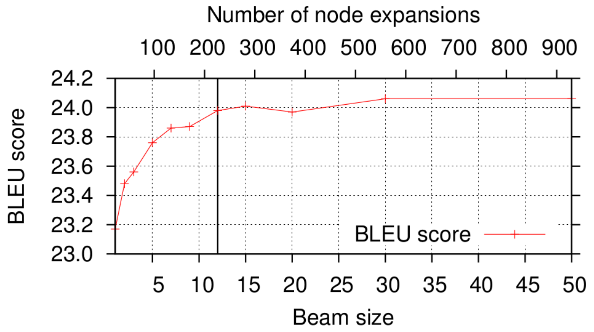

Figure 1: Performance with NPLM over beam size on English-German *news-test2015*. A beam of 12 corresponds to row 15 in Tab. 3.

Determinisation	Minimisation	Weight pushing	Sentences per second
✓			2.51
✓	✓		1.57
✓	✓	✓	1.47

Table 4: Time for lattice preprocessing operations on English-German *news-test2015*.

baselines, although lattice rescoring gives the best results (row 9 vs. row 15). Lattice rescoring with SGNMT also uses far fewer node expansions per sentence. We report n-best rescoring speeds for rescoring each hypothesis separately, and a depth-first (DFS) scheme that efficiently traverses the n-best lists. Both these techniques are very slow compared to lattice rescoring. Fig. 1 shows that we can reduce the beam size from 12 to 5 with only a minor drop in BLEU. This is nearly 100 times faster than DFS over the 1000-best list.

Cost of Lattice Preprocessing As described in Sec. 2.1, we applied determinisation, minimisation, and weight pushing to the Hiero lattices in order to work with probabilities. Tab. 4 shows that those operations are generally fast[4].

Lattice Size For previous experiments we set the Hiero pruning parameters such that lattices had 8,510 nodes on average. Fig. 2 plots the BLEU score over the lattice size. We find that SGNMT works well on lattices of moderate or large size, but pruning lattices too heavily has a negative effect as they are then too similar to Hiero first best hypotheses. We note that lattice rescoring involves nearly as many node expansions as unconstrained NMT decoding. This confirms that the lattices at 8,510 nodes are already large enough for SGNMT.

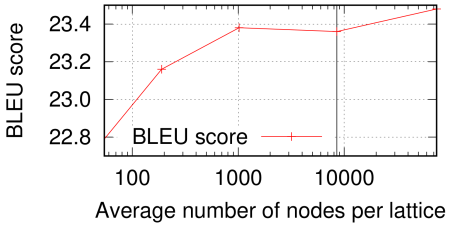

Figure 2: SGNMT performance over lattice size on English-German *news-test2015*. 8,510 nodes per lattice corresponds to row 14 in Tab. 3.

Local Softmax In SGNMT decoding we have the option of normalising the NMT translation probabilities over the words on outgoing words from each state rather than over the full 50,000 words translation vocabulary. There are ~4.5 arcs per state in our En-De'14 lattices, and so avoiding the full softmax could cause significant computational savings. We find this leads to only a modest 0.5 BLEU degradation: 21.45 BLEU in En-De'14, compared to 21.87 BLEU using NMT probabilities computed over the full vocabulary.

Modelling Errors vs. Search Errors In our En-De'14 experiments with $\lambda_{Hiero} = 0$ we find that constraining the NMT decoder to the Hiero lattices yields translation hypotheses with much lower NMT probabilities than unconstrained BASIC NMT decoding: under the NMT model, NMT hypotheses are 8,300 times more likely (median) than NMT-HIERO hypotheses. We conclude (tentatively) that BASIC NMT is not suffering only from search errors, but rather that NMT-HIERO discards some hypotheses ranked highly by the NMT model but lower in the evaluation metric.

4 Conclusion

We have demonstrated a viable approach to Syntactically Guided Neural Machine Translation formulated to exploit the rich, structured search space generated by Hiero and the long-context translation scores of NMT. SGNMT does not suffer from the severe limitation in vocabulary size of basic NMT and avoids any difficulty of extending distributed word representations to new vocabulary items not seen in training data.

Acknowledgements

This work was supported in part by the U.K. Engineering and Physical Sciences Research Council (EPSRC grant EP/L027623/1).

[4] Testing environment: Ubuntu 14.04, Linux 3.13.0, single Intel® Xeon® X5650 CPU at 2.67 GHz

References

Cyril Allauzen, Michael Riley, Johan Schalkwyk, Wojciech Skut, and Mehryar Mohri. 2007. OpenFst: A general and efficient weighted finite-state transducer library. In *Implementation and Application of Automata*, pages 11–23. Springer.

Cyril Allauzen, Bill Byrne, de Adrià Gispert, Gonzalo Iglesias, and Michael Riley. 2014. Pushdown automata in statistical machine translation. *Volume 40, Issue 3 - September 2014*, pages 687–723.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *ICLR*.

Frédéric Bastien, Pascal Lamblin, Razvan Pascanu, James Bergstra, Ian Goodfellow, Arnaud Bergeron, Nicolas Bouchard, David Warde-Farley, and Yoshua Bengio. 2012. Theano: new features and speed improvements. In *Deep Learning and Unsupervised Feature Learning NIPS 2012 Workshop*.

David Chiang, Kevin Knight, and Wei Wang. 2009. 11,001 new features for statistical machine translation. In *ACL*, pages 218–226.

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

Rohan Chitnis and John DeNero. 2015. Variable-length word encodings for neural translation models. In *EMNLP*, pages 2088–2093.

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *EMNLP*.

Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th International Conference on Machine Learning*, pages 160–167. ACM.

Adrià de Gispert, Gonzalo Iglesias, Graeme Blackwood, Eduardo R Banga, and William Byrne. 2010. Hierarchical phrase-based translation with weighted finite-state transducers and shallow-n grammars. *Computational Linguistics*, 36(3):505–533.

Ian Goodfellow, David Warde-farley, Mehdi Mirza, Aaron Courville, and Yoshua Bengio. 2013. Maxout networks. In *ICML*, pages 1319–1327.

Kenneth Heafield, Ivan Pouzyrevsky, Jonathan H. Clark, and Philipp Koehn. 2013. Scalable modified Kneser-Ney language model estimation. In *ACL*, pages 690–696.

Gonzalo Iglesias, Adrià de Gispert, Eduardo R Banga, and William Byrne. 2009. Hierarchical phrase-based translation with weighted finite state transducers. In *NAACL-HLT*, pages 433–441.

Gonzalo Iglesias, Cyril Allauzen, William Byrne, Adrià de Gispert, and Michael Riley. 2011. Hierarchical phrase-based translation representations. In *EMNLP*, pages 1373–1383.

Sébastien Jean, Kyunghyun Cho, Roland Memisevic, and Yoshua Bengio. 2015a. On using very large target vocabulary for neural machine translation. In *ACL*, pages 1–10.

Sébastien Jean, Orhan Firat, Kyunghyun Cho, Roland Memisevic, and Yoshua Bengio. 2015b. Montreal neural machine translation systems for WMT15. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 134–140.

Nal Kalchbrenner and Phil Blunsom. 2013. Recurrent continuous translation models. In *EMNLP*, page 413.

Jimmy Lin and Chris Dyer. 2010. *Data-intensive text processing with MapReduce*. Morgan &Claypool.

Minh-Thang Luong, Ilya Sutskever, Quoc V Le, Oriol Vinyals, and Wojciech Zaremba. 2015. Addressing the rare word problem in neural machine translation. In *ACL*.

Wolfgang Macherey, Franz Josef Och, Ignacio Thayer, and Jakob Uszkoreit. 2008. Lattice-based minimum error rate training for statistical machine translation. In *EMNLP*, pages 725–734.

Yuval Marton and Philip Resnik. 2008. Soft syntactic constraints for hierarchical phrased-based translation. In *ACL*, pages 1003–1011.

Mehryar Mohri and Michael Riley. 2001. A weight pushing algorithm for large vocabulary speech recognition. In *Interspeech*, pages 1603–1606.

Mehryar Mohri, Fernando Pereira, and Michael Riley. 2002. Weighted finite-state transducers in speech recognition. *Computer Speech and Language*, 16(1).

Graham Neubig, Makoto Morishita, and Satoshi Nakamura. 2015. Neural reranking improves subjective quality of machine translation: NAIST at WAT2015. In *Workshop on Asian Translation*, pages 35–41.

Holger Schwenk. 2014. Universit du Maine. `http://www-lium.univ-lemans.fr/~schwenk/nnmt-shared-task/`. [Online; accessed 1-March-2016].

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*.

Nakatani Shuyo. 2010. Language detection library for Java. `http://code.google.com/p/language-detection/`. [Online; accessed 1-March-2016].

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems*, pages 3104–3112.

Zhaopeng Tu, Zhengdong Lu, Yang Liu, Xiaohua Liu, and Hang Li. 2016. Coverage-based neural machine translation. *arXiv preprint arXiv:1601.04811*.

Bart van Merriënboer, Dzmitry Bahdanau, Vincent Dumoulin, Dmitriy Serdyuk, David Warde-Farley, Jan Chorowski, and Yoshua Bengio. 2015. Blocks and fuel: Frameworks for deep learning. *arXiv preprint arXiv:1506.00619*.

Ashish Vaswani, Yinggong Zhao, Victoria Fossum, and David Chiang. 2013. Decoding with large-scale neural language models improves translation. In *EMNLP*, pages 1387–1392.

Ashish Venugopal, Andreas Zollmann, Noah A. Smith, and Stephan Vogel. 2009. Preference grammars: Softening syntactic constraints to improve statistical machine translation. In *NAACL-HLT*, pages 236–244.