Extending hybrid word-character neural machine translation with multi-task learning of morphological analysis

Stig-Arne Grönroos and **Sami Virpioja** and **Mikko Kurimo**
stig-arne.gronroos@aalto.fi
Department of Signal Processing and Acoustics, Aalto University, Finland

Abstract

This article describes the Aalto University entry to the English-to-Finnish news translation shared task in WMT 2017. Our system is an open vocabulary neural machine translation (NMT) system, adapted to the needs of a morphologically complex target language. The main contributions of this paper are 1) implicitly incorporating morphological information to NMT through multi-task learning, 2) adding an attention mechanism to the character-level decoder, combined with character segmentation of names, and 3) a new overattending penalty to beam search.

1 Introduction

The rich inflection, derivation and compounding in synthetic languages can result in very large vocabularies. In statistical machine translation (SMT) large vocabularies cause sparsity issues. While continuous space representations make neural machine translation (NMT) more robust towards such sparsity, it suffers from a different set of problems related to large vocabularies. A large vocabulary bloats memory and computation requirements, while still leaving the problem of out-of-vocabulary words unsolved.

Subword vocabularies have been proposed as a solution. While the benefits of using subwords in SMT have been at best moderate (Virpioja et al., 2007; Fishel and Kirik, 2010; Grönroos et al., 2015), subword decoding has become popular in NMT (Sennrich et al., 2015). A subword vocabulary of a moderate size ensures full coverage of an open vocabulary. The downside is an increase in the length of the input and output sequences. Long sequences cause a large increase in computation time, especially for architectures using the attention mechanism.

An alternative approach is the hybrid word-character decoder presented by Luong and Manning (2016). In the hybrid decoder, a word level decoder outputs frequent words as they are, while replacing infrequent words with a special <UNK> symbol. A second character-level decoder then expands these <UNK> symbols into surface forms.

In addition to providing moderate length of input and output sequences together with an open vocabulary, the hybrid word-character decoder makes it simple to use labels based on the level of words, provided for example by morphological analyzers and parsers. In SMT, such tools are typically used via factored translation models (Koehn and Hoang, 2007). Factored translation has also been successfully applied in NMT. For example, Sennrich and Haddow (2016) augment the source words with four additional factors: PoS, lemma, dependency label and subwords. García-Martínez et al. (2016) use a decomposed generation process, in which they first output lemma, PoS, tense, person, gender, and number, from which the surface form is generated using a rule-based morphological analyzer.

Neural machine translation provides another way to utilize external annotations, multi-task learning (MTL). MTL is a well established machine learning approach that aims at improving the generalization performance of a task using other related tasks (Caruana, 1998). For example, Luong et al. (2016) use autoencoding, parsing, and caption generation as auxiliary tasks to improve English-to-German translation. Eriguchi et al. (2017) combine NMT with a Recurrent Neural Network Grammar. The system learns to parse the target language as an auxiliary task when translating into English.

We propose an MTL approach inspired by factored translation. The output of a morphological analyzer for the target sentence is used as an auxiliary prediction target, while sharing network parameters to a larger extent than in the approach of Luong et al. (2016).

This approach has two advantages over factored models. When training a system using factored output, embedded gold standard labels are given as input to the decoder. During translation gold standard labels are not available, and predicted labels are instead fed back in. The confidence of the predictions is not accounted for when feeding back the labels. This might worsen the problems caused by exposure bias, i.e., the mismatch between training and inference (Ranzato et al., 2016). If factored input is used, the external labeling tools need to be included also in the translation pipeline. In MTL such tools are only necessary during training.

In terms of computational cost, a factored model needs to predict the auxiliary labels also during translation, slowing down inference and complicating the beam search. A factored model might also need to use a larger beam to avoid hypotheses with the same surface form but different labels from crowding out more diverse hypotheses. In MTL, the auxiliary tasks are only performed during training, and no changes need to be made to the inference.

The main contributions of this paper are combining word-level labels from morphological analysis with a hybrid word-character decoder, and adding an attention mechanism to the character-level decoder. We also propose a new overattending penalty to the beam search.

2 Neural machine translation

Neural machine translation (NMT) is a framework for machine translation that uses a single neural network trained end-to-end. The recently proposed encoder-decoder network with attention mechanism (Bahdanau et al., 2014) has become accepted as the current standard in NMT.

The first part of the network, the encoder, reads a source sentence $\boldsymbol{x}$ and encodes it as a sequence of hidden states $s = (s_1, s_2, \dots, s_N)$. The encoder is often implemented as a bidirectional recurrent network with long short-term memory units (bi-LSTM), in which case each hidden state is the concatenation of a state from the forward and backward encoders.

The last part of the network, the decoder, is implemented as a conditional recurrent language model which models the probability of the target sentence $\boldsymbol{y}$ as

$$\begin{aligned}\log p(\boldsymbol{y} \mid \boldsymbol{x}) &= \sum_t \log p(y_t \mid \boldsymbol{y}_{<t}, \boldsymbol{x}) \\ &= \sum_t \log p(y_t \mid h_t, c_t). \qquad (1)\end{aligned}$$

The encoder and decoder are linked by the attention mechanism. At each timestep, the attention mechanism computes a context vector c_t as a weighted average of the encoder hidden states s. The weights $a_{t,i}$ are determined by a layer that takes as input the current decoder hidden state h_t and each of the vectors s_i in turn.

$$\begin{aligned}a_{t,i}(h, s) &= \frac{\exp(\text{align}(\boldsymbol{h}_t, \boldsymbol{s}_i))}{\sum_j \exp(\text{align}(\boldsymbol{h}_t, \boldsymbol{s}_j))} \\ \text{align}(\boldsymbol{h}_t, \boldsymbol{s}_i) &= \boldsymbol{v}_a^\top \tanh(W_a[\boldsymbol{h}_t; \boldsymbol{s}_i]) \qquad (2)\end{aligned}$$

In effect, at each timestep the attention mechanism scans the entire source to decide which parts are relevant to focus on when generating the next output symbol.

Luong and Manning (2016) extend the word-level encoder-decoder model by adding character-level processing of rare words. On the encoder side, word embeddings for rare source words are produced by a character-level encoder, instead of using a universal <UNK> embedding. The hybrid model ensures an open vocabulary, while keeping the attended sequence shorter than using characters or subwords.

On the decoder side, the word-level decoder outputs <UNK> for rare words, while storing the decoder hidden state at that timestep. A separate character-level decoder expands these tokens into the surface form. The character-level encoder and decoder can be trained jointly with the word-level components, by backpropagating end-to-end.

In separate-path initialization of the character-level decoder, the word-level LSTM output h is not used to seed the character-level decoder, but instead a counterpart vector $\breve{h}$ is calculated as

$$\breve{h}_t = \tanh(\breve{W}[c_t; h_t])$$

3 System description

Our system is based on the open-source Helsinki Neural Machine Translation (HNMT) software[1].

[1]Available from https://github.com/robertostling/hnmt .

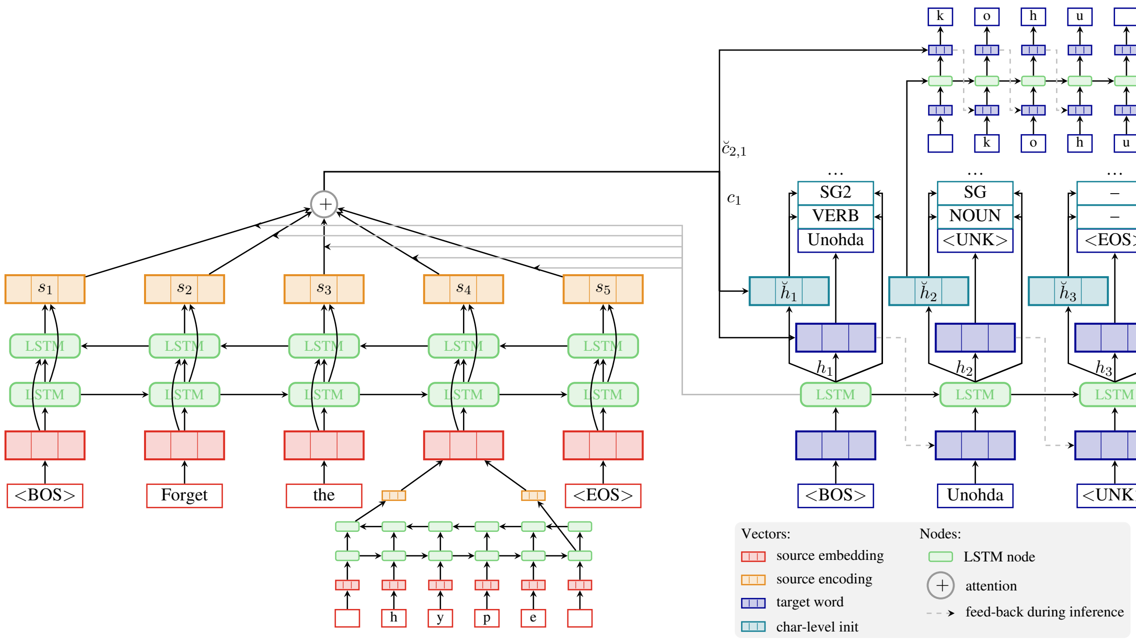

Figure 1: Our neural network architecture. In the example, "Forget the hype" is translated into "Unohda kohu". On the left side, the hybrid word-character encoder, using bi-LSTM for both levels. On the lower right side, the word-level attentional LSTM decoder, which predicts both word tokens and auxiliary labels. Above it, the predicted <UNK> is expanded by the attentional character-level decoder. For clarity, attention is only drawn for the first timestep of each decoder.

We extend[2] HNMT with a hybrid word-character decoder, multi-task learning, and improved beam search. An overview of the neural network architecture can be seen in Figure 1.

Hybrid encoder-decoder. HNMT implements a hybrid word-character encoder. Instead of the two-level unidirectional LSTM character-level encoders of Luong and Manning (2016), bi-LSTM encoders are used. The embedding for rare words is the concatenation of the last states of the forward and backward character-level encoders.

We extend HNMT with a hybrid word-character decoder, using separate path initialization of the character-level decoder. We also add an attention mechanism to the character-level decoder, yielding the character-level context vector $\breve{c}_{t,t_c}$. The attended sequence is the same as for the word-level decoder: the word-level encoding s of the source sentence. To make it possible for the attentional character-level decoder to copy or transcribe on a subword-level, we perform character segmentation preprocessing on capitalized input words (after truecasing). The segmentation is described in Section 4.

Multi-task learning. The main task is translation into the target language surface form, while the auxiliary tasks consist of predicting the output of the FinnPos morphological analyzer for the target sentence. The auxiliary tasks provide additional supervision signals that can help the model learn grammar and morphology. The tasks share parameters more closely than the one-to-many multi-task learning setting defined by Luong et al. (2016). In addition to sharing the encoder, all parts of the word level decoder except the final feed-forward prediction layers are shared. A potential downside compared to using a separate decoder is that the label sequence must be of the same length and synchronous with the surface sequence. This tightly shared MTL matches perfectly with the hybrid word-character decoder, as the labeling is on the level of words. The work-around of repeating labels to match the length of a subword sequence was not explored in this work.

In MTL, the supervision from the labels is softer than when using a factored model. Uncertain labels could be ignored, by limiting the task to sentences with high-confidence labels. We did not use this opportunity, as FinnPos labels every input sentence, and does not provide confidence estimates. As all our data $\mathcal{D}$ is labeled, we control the influence of the auxiliary task using a multiplica-

[2] Our fork available from https://github.com/Waino/hnmt .

tive weight on part of the cost function, instead of the minibatch mixing ratio used by Luong et al. (2016).

We train the whole model jointly to maximize

$$E_{(\boldsymbol{x},\boldsymbol{y},\boldsymbol{a})\in\mathcal{D}}[\log p(\boldsymbol{y},\boldsymbol{a}\,|\,\boldsymbol{x})]$$

where $\boldsymbol{a}$ are the labels: the cluster id of the lemma, the rounded log-frequency of the lemma, the PoS, and 5 morphological tags: number, case, person, mood, and tense. Each label is independently predicted from the concatenation of h and $\breve{h}$.

Beam search scoring function. We use beam search during decoding to find the optimal translation sequence $\boldsymbol{y}$. Instead of directly maximizing the probability, we maximize a score function $s(\boldsymbol{y},\boldsymbol{x})$, designed to alleviate two known issues in NMT: overtranslation and undertranslation.

Undertranslation is reduced by adding length normalization (lp) and a coverage penalty (cp), following Wu et al. (2016).

Unlike undertranslation, overtranslation is to some extent inherently reduced by the monotonically increasing generation log-probabilities. However, the inherent cost is not enough, leading us to add a penalty for overattending a source token (oap). The penalty is applied if the most attended source word has sum attention over 1.0. We use the maximum function instead of sum, in order not to increase the strength of the penalty for long input sentences. The overattending penalty is monotonically increasing, which enables us to include it when pruning active hypotheses.

The overattending penalty is not suitable if the decoder uses smaller units than the output of the encoder. Repeated attention is required if the decoder must output several subwords for each source token.

The scoring function is

$$\begin{aligned} s(\boldsymbol{y},\boldsymbol{x}) = &-\log\big(p(\boldsymbol{y}\,|\,\boldsymbol{x})\big) + \mathrm{lp}(\boldsymbol{y}) \\ &+ \mathrm{cp}(\boldsymbol{y},\boldsymbol{x}) + \mathrm{oap}(\boldsymbol{y},\boldsymbol{x}), \end{aligned} \tag{3}$$

where

$$\mathrm{lp}(\boldsymbol{y}) = \frac{(|\boldsymbol{y}|+\lambda)^\alpha}{(1+\lambda)^\alpha} \tag{4}$$

$$\mathrm{cp}(\boldsymbol{y},\boldsymbol{x}) = \beta\sum_{i=1}^{|\boldsymbol{x}|}\log\big(\min(\sum_{j=1}^{|\boldsymbol{y}|}a_{ij}, 1.0)\big) \tag{5}$$

$$\mathrm{oap}(\boldsymbol{y},\boldsymbol{x}) = -\gamma\max\big(\max_{i=1}^{|\boldsymbol{x}|}(\sum_{j=1}^{|\boldsymbol{y}|}a_{ij} - 1.0), 0.0\big) \tag{6}$$

The parameters α, β, γ, and λ control the strengths of the penalties.

Pruning in beam search. We use three types of pruning in the beam search.

First, at each step, for each hypothesis to be extended, we prune the list of candidates for the next symbol based on local probability, to only keep $beam\_width + 1$ candidates. This pruning improves speed without affecting the output.

Second, after at least one hypothesis has been completed, we keep track of the current best normalized score. This allows pruning active hypotheses by comparing their partially normalized score against the best normalized score, with adjustable pruning margin. The partially normalized score is calculated as the sum of the monotonically increasing parts of the scoring function

$$-\log\big(p(\boldsymbol{y}\,|\,\boldsymbol{x})\big) + \mathrm{oap}(\boldsymbol{y},\boldsymbol{x})$$

This pruning may affect the output by removing a hypothesis with a poor early score that could have improved later. To gain a speed-up, it is necessary to prune active hypotheses: limiting pruning to completed hypotheses cannot reduce the number of hypotheses in early stages, and thus cannot result in early clearing of the beam.

Completed hypotheses are moved from the beam to a separate heap. This clears out room in the beam for active hypotheses, but also means that the pruning of active hypotheses becomes essential for early stopping of the beam search.

The third type of pruning is applied to the heap of completed hypotheses based on normalized score, to only keep n best hypotheses. This pruning conserves memory and does not affect the ordering of the results.

4 Data

Our system participates in the constrained condition of the WMT shared task. As training data, we used the Europarl-v8, Rapid and Wikititles corpora, extended with backtranslated monolingual data, resulting in 6 091 184 parallel sentence pairs after cleaning. The backtranslated sentences were from the news.2014.fi corpus, translated with a PBSMT model, trained with WMT16 constrained settings. Based on initial experiments we decided to use the full backtranslated set, for a ratio of ca 60% backtranslated to 40% parallel data, instead of subsampling to balance the ratio.

	newstest2016AB			newstest2017			
Configuration	chrF-1	chrF-2	BLEU	chrF-1	chrF-2	BLEU	TER
Hybrid decoder with MTL, ensemble of 4	56.79	**55.60**	21.46	57.30	**55.96**	20.28	**.673**
+ repetition removal	**57.07**	55.59	**21.55**	**57.57**	55.92	**20.31**	–
FlatCat subword decoder, ensemble of 4	55.77	55.41	20.01	54.10	53.98	17.15	.750
Hybrid decoder with MTL, single model	54.69	53.43	18.60	55.17	53.87	17.84	–

Table 1: Results of automatic evaluation. BLEU and chrF scores are percentages. TER from http://matrix.statmt.org/matrix/systems_list/1871?metric_id=2.

	newstest2016AB		
Configuration	chrF-1	chrF-2	BLEU
Hybrid decoder with MTL	56.79	55.60	21.46
No morphological tags	55.97	55.20	19.83
No log frequency	55.49	54.26	19.47
No clustered lemma	55.23	53.65	19.37
No PoS-tags	55.05	53.73	19.29
No multi-task learning	54.91	53.48	19.43
No character attention & name segmentation	52.12	50.80	17.16
No length penalty	56.68	55.52	21.35
No overattending penalty	56.68	55.53	21.33
No coverage penalty	56.43	54.93	20.97
No penalties	55.90	54.21	20.45

Table 2: Results of ablation experiments. All runs are ensembles of 4, to reduce variability.

Data preprocessing consists of filtering too long sentences, normalizing misencoded data, normalizing punctuation, deduplication, tokenization, statistical truecasing, filtering of untranslated sentences, and character segmentation of names on the source side.

Segmenting names into characters, when combined with attention on the character level, allows copying or transliteration on a character-to-character basis. It is applied using a rough heuristic: we segment any token longer than one character beginning with an upper case letter or digit. All segmented characters are marked using reserved symbols. The first and last characters of the sequence have distinct symbols separating them from word-internal characters.

The filtering of untranslated sentences was also performed using a rough heuristic, by filtering any sentences containing certain common English contractions and clitics that do not occur in Finnish. The target side training data, especially Europarl, contains hundreds of sentences with English phrases. A typical reason is discussions on the wording of English-language documents being drafted. The filtering was an attempt to alleviate a failure mode in which the system would instead of translating attempt (and fail) to output the English source.

A parallel corpus augmented with gold-standard labels for MTL is not available. We tag the target side of the parallel corpus using the statistical tagger FinnPos (Silfverberg et al., 2016). The resulting labels are noisy, but nonetheless provide supervision for the morphological analysis task.

We postprocess the output of FinnPos. The morpheme tag sequence is split, and tags are grouped by type. FinnPos lemmas are noisy, containing many remaining affixes and other mislemmatizations. We collapse numbers into a single number symbol, remove special characters, and cluster the remaining lemmas into 10 000 clusters with word2vec (Mikolov et al., 2013).

5 Training details

We use the following parameters for the network: weight of auxiliary task between 0.001 and 0.75, 64 dimensional character embeddings, 256 dimensional word embeddings, 128 dimensional aux embeddings, 2*256 dimensional encoder state, 1024 dimensional word decoder state, 1024 dimensional character decoder state, 256 dimensional attention, everything except 25k most frequent source words embedded by character level encoder, 50k most frequent target words output by word level decoder, 10k overlap between word level and character level vocabularies during training.

For training, we use Adam with initial learning rate 0.001 and gradient norm clipped to 5.0.

The systems have been tuned towards characterF-1.0 (Popovic, 2015, 2016). We optimize the beam search parameters, using a grid search. The optimal parameters were α 0.012,

β 0.3, γ 0.2, λ 3, pruning margin 1.4, and weight 0.8 for the character-level cost.

We use an ensemble procedure, in which the combined prediction is computed as the mean after the softmax layer of the predictions of 4 models. The primary system uses systems from 4 runs with different weights for the auxiliary task. The systems trained for comparison—a subword system based on Morfessor FlatCat and the systems in ablation experiments—were ensembled using 4 save points from a single run.

To include an example of subword NMT, we also submit our FlatCat system. As preprocessing, the target side has been segmented using Morfessor FlatCat (Grönroos et al., 2014), which was tuned to produce a subword lexicon of approximately 60k symbols. Segmenting names into characters is applied in addition to the FlatCat segmentation. The FlatCat segmented system uses WMT 2016 data only, i.e., omits the Rapid corpus.

The FlatCat subword system uses the standard HNMT decoder. It uses neither the hybrid word-character decoder nor MTL. We did however use the improved beam search with penalties.

6 Results

We evaluate the systems using characterF with β set to 1.0 and 2.0, and cased BLEU using the `mteval-v13a.pl` script. We also include Translation Error Rate (TER) results for the submitted systems. Our primary system has the best TER score of all participants.

As the development test set we use both reference translations of the newstest 2016 set. Table 1 shows the submitted ensemble systems, and the best single model for our primary system. As our system has a tendency to repeat certain words, we also evaluate the primary system after a post-processing step in which consecutive repetitions are removed.

We perform ablation experiments for all new components in our system, by removing each of them separately (non-cumulative effect). Results are shown in Table 2.

All added components were beneficial. The largest improvement, +4.3 BLEU, comes from the attention mechanism in the character decoder, combined with segmenting names into characters.

Multi-task learning improves BLEU by +2.03. Not all auxiliary labels are equally important. PoS tags (+2.17 BLEU) and clustered lemmas (+2.09 BLEU) perform above average, and removing either of them yields worse BLEU than not using MTL at all. The results of both characterF measures differ in this, ranking not using MTL as worse than all the partial MTL variants.

The overattending penalty to the beam search gives a much more modest gain of +0.13 BLEU. The coverage penalty is the most important of the beam search penalties. In total, the beam search heuristics yield an improvement of +1.01 BLEU.

In the human evaluation, our primary system was ranked in the second of five clusters (tied 3rd to 5th place).

7 Discussion

All our added components improved the translation quality.

The largest improvement comes from the modifications intended to enable character-to-character copying: segmenting names into characters and character-level attention. However, the simple heuristic used for selecting words to segment can make translation more difficult in some cases, e.g. the names of institutions are typically capitalized, but translated on a term level. Replacing the heuristic with named-entity recognition or other more advanced methods is left for future work.

A common type of error made by our system is overtranslation through repetition. A possible explanation for the effect is the way that the levels of the hybrid word-character decoder are connected. There is no connection from the character level back to the word level. The surface forms generated by the character-level decoder are conditionally independent given the word-level hidden states, which can be similar to the states at adjacent time steps. The word-level decoder must decide on the number of words in an expression, which is a difficult task if the proportion of <UNK> tokens becomes large. The overattending penalty is only partially successful at reducing the repetition, and increasing the penalty weight deteriorates overall performance before eliminating the problem.

8 Conclusion

Our results show that translation into a morphologically complex language can be improved using word-level labels from morphological analysis combined with a hybrid word-character decoder. Adding an attention mechanism to the character decoder yields a large quality improvement.

Acknowledgements

This research has been supported by the Academy of Finland under the Finnish Centre of Excellence Program 2012–2017 (grant n°251170). Computer resources within the Aalto University School of Science "Science-IT" project were used. We wish to thank Jörg Tiedemann for the useful discussions and for sharing his backtranslations.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. In *ICLR15*.

Rich Caruana. 1998. Multitask learning. In *Learning to learn*, Springer, pages 95–133.

Akiko Eriguchi, Yoshimasa Tsuruoka, and Kyunghyun Cho. 2017. Learning to parse and translate improves neural machine translation. *arXiv preprint arXiv:1702.03525* .

Mark Fishel and Harri Kirik. 2010. Linguistically motivated unsupervised segmentation for machine translation. In *Language Resources and Evaluation (LREC)*. Valletta, Malta.

Mercedes García-Martínez, Loïc Barrault, and Fethi Bougares. 2016. Factored neural machine translation. *arXiv preprint arXiv:1609.04621* .

Stig-Arne Grönroos, Sami Virpioja, and Mikko Kurimo. 2015. Tuning phrase-based segmented translation for a morphologically complex target language. In *WMT15*. Association for Computational Linguistics, Lisbon, Portugal, pages 105–111.

Stig-Arne Grönroos, Sami Virpioja, Peter Smit, and Mikko Kurimo. 2014. Morfessor FlatCat: An HMM-based method for unsupervised and semi-supervised learning of morphology. In *COLING14*. Association for Computational Linguistics, pages 1177–1185.

Philipp Koehn and Hieu Hoang. 2007. Factored translation models. In *EMNLP-CoNLL*. pages 868–876.

Minh-Thang Luong, Quoc V Le, Ilya Sutskever, Oriol Vinyals, and Lukasz Kaiser. 2016. Multi-task sequence to sequence learning. In *ICLR*.

Minh-Thang Luong and Christopher D. Manning. 2016. Achieving open vocabulary neural machine translation with hybrid word-character models. In *ACL16*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems (NIPS)*. Lake Tahoe, NV, USA, pages 3111–3119.

Maja Popovic. 2015. chrF: character n-gram F-score for automatic MT evaluation. In *WMT15*. pages 392–395.

Maja Popovic. 2016. chrF deconstructed: β parameters and n-gram weights. In *WMT16*.

Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. 2016. Sequence level training with recurrent neural networks. In *International Conference on Learning Representations (ICLR)*.

Rico Sennrich and Barry Haddow. 2016. Linguistic input features improve neural machine translation. In *WMT16*. Association for Computational Linguistics, pages 83–91.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Neural machine translation of rare words with subword units. In *ACL16*.

Miikka Silfverberg, Teemu Ruokolainen, Krister Lindén, and Mikko Kurimo. 2016. FinnPos: an open-source morphological tagging and lemmatization toolkit for Finnish. *Language Resources and Evaluation (LREC)* 50(4):863–878.

Sami Virpioja, Jaakko J Väyrynen, Mathias Creutz, and Markus Sadeniemi. 2007. Morphology-aware statistical machine translation based on morphs induced in an unsupervised manner. *Machine Translation Summit XI* 2007:491–498.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144* .