Linguistic Input Features Improve Neural Machine Translation

Rico Sennrich and **Barry Haddow**
School of Informatics, University of Edinburgh
rico.sennrich@ed.ac.uk, bhaddow@inf.ed.ac.uk

Abstract

Neural machine translation has recently achieved impressive results, while using little in the way of external linguistic information. In this paper we show that the strong learning capability of neural MT models does not make linguistic features redundant; they can be easily incorporated to provide further improvements in performance. We generalize the embedding layer of the encoder in the attentional encoder–decoder architecture to support the inclusion of arbitrary features, in addition to the baseline word feature. We add morphological features, part-of-speech tags, and syntactic dependency labels as input features to English↔German and English→Romanian neural machine translation systems. In experiments on WMT16 training and test sets, we find that linguistic input features improve model quality according to three metrics: perplexity, BLEU and CHRF3. An open-source implementation of our neural MT system is available[1], as are sample files and configurations[2].

1 Introduction

Neural machine translation has recently achieved impressive results (Bahdanau et al., 2015; Jean et al., 2015), while learning from raw, sentence-aligned parallel text and using little in the way of external linguistic information.[3] However, we hypothesize that various levels of linguistic annotation can be valuable for neural machine translation. Lemmatisation can reduce data sparseness, and allow inflectional variants of the same word to explicitly share a representation in the model. Other types of annotation, such as parts-of-speech (POS) or syntactic dependency labels, can help in disambiguation. In this paper we investigate whether linguistic information is beneficial to neural translation models, or whether their strong learning capability makes explicit linguistic features redundant.

Let us motivate the use of linguistic features using examples of actual translation errors by neural MT systems. In translation out of English, one problem is that the same surface word form may be shared between several word types, due to homonymy or word formation processes such as conversion. For instance, *close* can be a verb, adjective, or noun, and these different meanings often have distinct translations into other languages. Consider the following English→German example:

1. *We thought a win like this might be close.*
2. *Wir dachten, dass ein solcher Sieg nah sein könnte.*
3. **Wir dachten, ein Sieg wie dieser könnte schließen.*

For the English source sentence in Example 1 (our translation in Example 2), a neural MT system (our baseline system from Section 4) mistranslates *close* as a verb, and produces the German verb *schließen* (Example 3), even though *close* is an adjective in this sentence, which has the German translation *nah*. Intuitively, part-of-speech annotation of the English input could disambiguate between verb, noun, and adjective meanings of *close*.

As a second example, consider the following German→English example:

4. *Gefährlich ist die Route aber dennoch .*
   dangerous is the route but still .

[1] https://github.com/rsennrich/nematus

[2] https://github.com/rsennrich/wmt16-scripts

[3] Linguistic tools are most commonly used in preprocessing, e.g. for Turkish segmentation (Gülçehre et al., 2015).

Proceedings of the First Conference on Machine Translation, Volume 1: Research Papers, pages 83–91,
Berlin, Germany, August 11-12, 2016. ©2016 Association for Computational Linguistics

5. *However the route is dangerous .*

6. **Dangerous is the route , however .*

German main clauses have a verb-second (V2) word order, whereas English word order is generally SVO. The German sentence (Example 4; English reference in Example 5) topicalizes the predicate *gefährlich* 'dangerous', putting the subject *die Route* 'the route' after the verb. Our baseline system (Example 6) retains the original word order, which is highly unusual in English, especially for prose in the news domain. A syntactic annotation of the source sentence could support the attentional encoder-decoder in learning which words in the German source to attend (and translate) first.

We will investigate the usefulness of linguistic features for the language pair German↔English, considering the following linguistic features:

- lemmas
- subword tags (see Section 3.2)
- morphological features
- POS tags
- dependency labels

The inclusion of lemmas is motivated by the hope for a better generalization over inflectional variants of the same word form. The other linguistic features are motivated by disambiguation, as discussed in our introductory examples.

2 Neural Machine Translation

We follow the neural machine translation architecture by Bahdanau et al. (2015), which we will briefly summarize here.

The neural machine translation system is implemented as an attentional encoder-decoder network with recurrent neural networks.

The encoder is a bidirectional neural network with gated recurrent units (Cho et al., 2014) that reads an input sequence $x = (x_1, ..., x_m)$ and calculates a forward sequence of hidden states $(\overrightarrow{h}_1, ..., \overrightarrow{h}_m)$, and a backward sequence $(\overleftarrow{h}_1, ..., \overleftarrow{h}_m)$. The hidden states $\overrightarrow{h}_j$ and $\overleftarrow{h}_j$ are concatenated to obtain the annotation vector h_j.

The decoder is a recurrent neural network that predicts a target sequence $y = (y_1, ..., y_n)$. Each word y_i is predicted based on a recurrent hidden state s_i, the previously predicted word y_{i-1}, and a context vector c_i. c_i is computed as a weighted sum of the annotations h_j. The weight of each annotation h_j is computed through an *alignment model* α_{ij}, which models the probability that y_i is aligned to x_j. The alignment model is a single-layer feedforward neural network that is learned jointly with the rest of the network through backpropagation.

A detailed description can be found in (Bahdanau et al., 2015), although our implementation is based on a slightly modified form of this architecture, released for the dl4mt tutorial[4]. Training is performed on a parallel corpus with stochastic gradient descent. For translation, a beam search with small beam size is employed.

2.1 Adding Input Features

Our main innovation over the standard encoder-decoder architecture is that we represent the encoder input as a combination of features (Alexandrescu and Kirchhoff, 2006).

We here show the equation for the forward states of the encoder (for the simple RNN case; consider (Bahdanau et al., 2015) for GRU):

$$\overrightarrow{h}_j = \tanh(\overrightarrow{W} E x_j + \overrightarrow{U}\,\overrightarrow{h}_{j-1}) \qquad (1)$$

where $E \in \mathbb{R}^{m \times K_x}$ is a word embedding matrix, $\overrightarrow{W} \in \mathbb{R}^{n \times m}$, $\overrightarrow{U} \in \mathbb{R}^{n \times n}$ are weight matrices, with m and n being the word embedding size and number of hidden units, respectively, and K_x being the vocabulary size of the source language.

We generalize this to an arbitrary number of features $|F|$:

$$\overrightarrow{h}_j = \tanh(\overrightarrow{W}(\mathop{\Big\|}_{k=1}^{|F|} E_k x_{jk}) + \overrightarrow{U}\,\overrightarrow{h}_{j-1}) \qquad (2)$$

where $\|$ is the vector concatenation, $E_k \in \mathbb{R}^{m_k \times K_k}$ are the feature embedding matrices, with $\sum_{k=1}^{|F|} m_k = m$, and K_k is the vocabulary size of the kth feature. In other words, we look up separate embedding vectors for each feature, which are then concatenated. The length of the concatenated vector matches the total embedding size, and all other parts of the model remain unchanged.

[4] https://github.com/nyu-dl/dl4mt-tutorial

3 Linguistic Input Features

Our generalized model of the previous section supports an arbitrary number of input features. In this paper, we will focus on a number of well-known linguistic features. Our main empirical question is if providing linguistic features to the encoder improves the translation quality of neural machine translation systems, or if the information emerges from training encoder-decoder models on raw text, making its inclusion via explicit features redundant. All linguistic features are predicted automatically; we use Stanford CoreNLP (Toutanova et al., 2003; Minnen et al., 2001; Chen and Manning, 2014) to annotate the English input for English→German, and ParZu (Sennrich et al., 2013) to annotate the German input for German→English. We here discuss the individual features in more detail.

3.1 Lemma

Using lemmas as input features guarantees sharing of information between word forms that share the same base form. In principle, neural models can learn that inflectional variants are semantically related, and represent them as similar points in the continuous vector space (Mikolov et al., 2013). However, while this has been demonstrated for high-frequency words, we expect that a lemmatized representation increases data efficiency; low-frequency variants may even be unknown to word-level models. With character- or subword-level models, it is unclear to what extent they can learn the similarity between low-frequency word forms that share a lemma, especially if the word forms are superficially dissimilar. Consider the following two German word forms, which share the lemma *liegen* 'lie':

- *liegt* 'lies' (3.p.sg. present)
- *läge* 'lay' (3.p.sg. subjunctive II)

The lemmatisers we use are based on finite-state methods, which ensures a large coverage, even for infrequent word forms. We use the Zmorge analyzer for German (Schmid et al., 2004; Sennrich and Kunz, 2014), and the lemmatiser in the Stanford CoreNLP toolkit for English (Minnen et al., 2001).

3.2 Subword Tags

In our experiments, we operate on the level of subwords to achieve open-vocabulary translation with a fixed symbol vocabulary, using a segmentation based on *byte-pair encoding* (BPE) (Sennrich et al., 2016c). We note that in BPE segmentation, some symbols are potentially ambiguous, and can either be a separate word, or a subword segment of a larger word. Also, text is represented as a sequence of subword units with no explicit word boundaries, but word boundaries are potentially helpful to learn which symbols to attend to, and when to forget information in the recurrent layers. We propose an annotation of subword structure similar to popular IOB format for chunking and named entity recognition, marking if a symbol in the text forms the beginning (B), inside (I), or end (E) of a word. A separate tag (O) is used if a symbol corresponds to the full word.

3.3 Morphological Features

For German→English, the parser annotates the German input with morphological features. Different word types have different sets of features – for instance, nouns have case, number and gender, while verbs have person, number, tense and aspect – and features may be underspecified. We treat the concatenation of all morphological features of a word, using a special symbol for underspecified features, as a string, and treat each such string as a separate feature value.

3.4 POS Tags and Dependency Labels

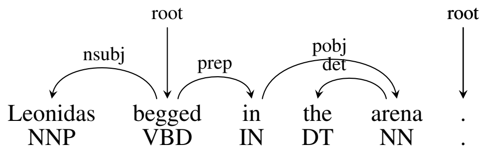
In our introductory examples, we motivated POS tags and dependency labels as possible disambiguators. Each word is associated with one POS tag, and one dependency label. The latter is the label of the edge connecting a word to its syntactic head, or 'ROOT' if the word has no syntactic head.

3.5 On Using Word-level Features in a Subword Model

We segment rare words into subword units using BPE. The subword tags encode the segmentation of words into subword units, and need no further modification. All other features are originally word-level features. To annotate the segmented source text with features, we copy the word's feature value to all its subword units. An example is shown in Figure 1.

4 Evaluation

We evaluate our systems on the WMT16 shared translation task English↔German. The parallel



words	Le:	oni:	das	beg:	ged	in	the	arena	.
lemmas	Leonidas	Leonidas	Leonidas	beg	beg	in	the	arena	.
subword tags	B	I	E	B	E	O	O	O	O
POS	NNP	NNP	NNP	VBD	VBD	IN	DT	NN	.
dep	nsubj	nsubj	nsubj	root	root	prep	det	pobj	root

Figure 1: Original dependency tree for sentence *Leonidas begged in the arena .*, and our feature representation after BPE segmentation.

training data consists of about 4.2 million sentence pairs.

To enable open-vocabulary translation, we encode words via joint BPE[5] (Sennrich et al., 2016c), learning 89 500 merge operations on the concatenation of the source and target side of the parallel training data. We use minibatches of size 80, a maximum sentence length of 50, word embeddings of size 500, and hidden layers of size 1024. We clip the gradient norm to 1.0 (Pascanu et al., 2013). We train the models with Adadelta (Zeiler, 2012), reshuffling the training corpus between epochs. We validate the model every 10 000 minibatches via BLEU and perplexity on a validation set (newstest2013).

For neural MT, perplexity is a useful measure of how well the model can predict a reference translation given the source sentence. Perplexity is thus a good indicator of whether input features provide any benefit to the models, and we report the best validation set perplexity of each experiment. To evaluate whether the features also increase translation performance, we report case-sensitive BLEU scores with mteval-13b.perl on two test sets, newstest2015 and newstest2016. We also report CHRF3 (Popović, 2015), a character n-gram F_3 score which was found to correlate well with human judgments, especially for translations out of English (Stanojević et al., 2015).[6] The two metrics may occasionally disagree, partly because they are highly sensitive to the length of the output. BLEU is precision-based, whereas CHRF3 considers both precision and recall, with a bias for recall. For BLEU, we also report whether differences between systems are statistically significant according to a bootstrap resampling significance test (Riezler and Maxwell, 2005).

	input vocabulary			embedding	
feature	EN	DE	model	all	single
subword tags	4	4	4	5	5
POS tags	46	54	54	10	10
morph. features	-	1400	1400	10	10
dependency labels	46	33	46	10	10
lemmas	800000	1500000	85000	115	167
words	78500	85000	85000	*	*

Table 1: Vocabulary size, and size of embedding layer of linguistic features, in system that includes all features, and contrastive experiments that add a single feature over the baseline. The embedding layer size of the word feature is set to bring the total size to 500.

We train models for about a week, and report results for an ensemble of the 4 last saved models (with models saved every 12 hours). The ensemble serves to smooth the variance between single models.

Decoding is performed with beam search with a beam size of 12.

To ensure that performance improvements are not simply due to an increase in the number of model parameters, we keep the total size of the embedding layer fixed to 500. Table 1 lists the embedding size we use for linguistic features – the embedding layer size of the word-level feature varies, and is set to bring the total embedding layer size to 500. If we include the lemma feature, we roughly split the embedding vector one-to-two between the lemma feature and the word feature. The table also shows the network vocabulary size; for all features except lemmas, we can represent all feature values in the network vocabulary – in the case of words, this is due to BPE segmentation. For lemmas, we choose the same vocabulary size as for words, replacing rare lemmas with a

[5] `https://github.com/rsennrich/subword-nmt`

[6] We use the re-implementation included with the subword code

special UNK symbol.

Sennrich et al. (2016b) report large gains from using monolingual in-domain training data, automatically back-translated into the source language to produce a synthetic parallel training corpus. We use the synthetic corpora produced in these experiments[7] (3.6–4.2 million sentence pairs), and we trained systems which include this data to compare against the state of the art. We note that our experiments with this data entail a syntactic annotation of automatically translated data, which may be a source of noise. For the systems with synthetic data, we double the training time to two weeks.

We also evaluate linguistic features for the lower-resourced translation direction English→Romanian, with 0.6 million sentence pairs of parallel training data, and 2.2 million sentence pairs of synthetic parallel data. We use the same linguistic features as for English→German. We follow Sennrich et al. (2016a) in the configuration, and use dropout for the English→Romanian systems. We drop out full words (both on the source and target side) with a probability of 0.1. For all other layers, the dropout probability is set to 0.2.

4.1 Results

Table 2 shows our main results for German→English, and English→German. The baseline system is a neural MT system with only one input feature, the (sub)words themselves. For both translation directions, linguistic features improve the best perplexity on the development data (47.3 → 46.2, and 54.9 → 52.9, respectively). For German→English, the linguistic features lead to an increase of 1.5 BLEU (31.4→32.9) and 0.5 CHRF3 (58.0 → 58.5), on the newstest2016 test set. For English→German, we observe improvements of 0.6 BLEU (27.8 → 28.4) and 1.2 CHRF3 (56.0 → 57.2).

To evaluate the effectiveness of different linguistic features in isolation, we performed contrastive experiments in which only a single feature was added to the baseline. Results are shown in Table 3. Unsurprisingly, the combination of all features (Table 2) gives the highest improvement, averaged over metrics and test sets, but most features are beneficial on their own. Subword tags give small improvements for English→German, but not for German→English. All other features outperform the baseline in terms of perplexity, and yield significant improvements in BLEU on at least one test set. The gain from different features is not fully cumulative; we note that the information encoded in different features overlaps. For instance, both the dependency labels and the morphological features encode the distinction between German subjects and accusative objects, the former through different labels (*subj* and *obja*), the latter through grammatical case (*nominative* and *accusative*).

We also evaluated adding linguistic features to a stronger baseline, which includes synthetic parallel training data. In addition, we compare our neural systems against phrase-based (PBSMT) and syntax-based (SBSMT) systems by (Williams et al., 2016), all of which make use of linguistic annotation on the source and/or target side. Results are shown in Table 4. For German→English, we observe similar improvements in the best development perplexity (45.2 → 44.1), test set BLEU (37.5→38.5) and CHRF3 (62.2 → 62.8). Our test set BLEU is on par to the best submitted system to this year's WMT 16 shared translation task, which is similar to our baseline MT system, but which also uses a right-to-left decoder for reranking (Sennrich et al., 2016a). We expect that linguistic input features and bidirectional decoding are orthogonal, and that we could obtain further improvements by combining the two.

For English→German, improvements in development set perplexity carry over (49.7 → 48.4), but we see only small, non-significant differences in BLEU and CHRF3. While we cannot clearly account for the discrepancy between perplexity and translation metrics, factors that potentially lower the usefulness of linguistic features in this setting are the stronger baseline, trained on more data, and the low robustness of linguistic tools in the annotation of the noisy, synthetic data sets. Both our baseline neural MT systems and the systems with linguistic features substantially outperform phrase-based and syntax-based systems for both translation directions.

In the previous tables, we have reported the best perplexity. To address the question about the randomness in perplexity, and whether the best perplexity just happened to be lower for the systems with linguistic features, we show perplexity on our development set as a function of training time

[7] The corpora are available at `http://statmt.org/rsennrich/wmt16_backtranslations/`

	German→English					English→German				
system	ppl ↓	BLEU ↑		chrF3 ↑		ppl ↓	BLEU ↑		chrF3 ↑	
	dev	test15	test16	test15	test16	dev	test15	test16	test15	test16
baseline	47.3	27.9	31.4	54.0	58.0	54.9	23.0	27.8	52.6	56.0
all features	46.2	28.7*	32.9*	54.8	58.5	52.9	23.8*	28.4*	53.9	57.2

Table 2: German↔English translation results: best perplexity on dev (newstest2013), and BLEU and chrF3 on test15 (newstest2015) and test16 (newstest2016). BLEU scores that are significantly different ($p < 0.05$) from respective baseline are marked with (*).

	German→English					English→German				
system	ppl ↓	BLEU ↑		chrF3 ↑		ppl ↓	BLEU ↑		chrF3 ↑	
	dev	test15	test16	test15	test16	dev	test15	test16	test15	test16
baseline	47.3	27.9	31.4	54.0	58.0	54.9	23.0	27.8	52.6	56.0
lemmas	47.1	28.4	32.3*	54.6	58.7	53.4	23.8*	28.5*	53.7	56.7
subword tags	47.3	27.7	31.5	54.0	58.1	54.7	23.6*	28.1	53.2	56.4
morph. features	47.1	28.2	32.4*	54.3	58.4	-	-	-	-	-
POS tags	46.9	28.1	32.4*	54.1	57.8	53.2	24.0*	28.9*	53.3	56.8
dependency labels	46.9	28.1	31.8*	54.2	58.3	54.0	23.4*	28.0	53.1	56.5

Table 3: Contrastive experiments with individual linguistic features: best perplexity on dev (newstest2013), and BLEU and chrF3 on test15 (newstest2015) and test16 (newstest2016). BLEU scores that are significantly different ($p < 0.05$) from respective baseline are marked with (*).

	German→English					English→German				
system	ppl ↓	BLEU ↑		chrF3 ↑		ppl ↓	BLEU ↑		chrF3 ↑	
	dev	test15	test16	test15	test16	dev	test15	test16	test15	test16
PBSMT (Williams et al., 2016)	-	29.9	35.1	56.2	60.9	-	23.7	28.4	52.6	56.6
SBSMT (Williams et al., 2016)	-	29.5	34.4	56.0	61.0	-	24.5	30.6	55.3	59.9
baseline	45.2	31.5	37.5	57.0	62.2	49.7	27.5	33.1	56.3	60.5
all features	44.1	32.1*	38.5*	57.5	62.8	48.4	27.1	33.2	56.5	60.6

Table 4: German↔English translation results with additional, synthetic training data: best perplexity on dev (newstest2013), and BLEU and chrF3 on test15 (newstest2015) and test16 (newstest2016). BLEU scores that are significantly different ($p < 0.05$) from respective baseline are marked with (*).

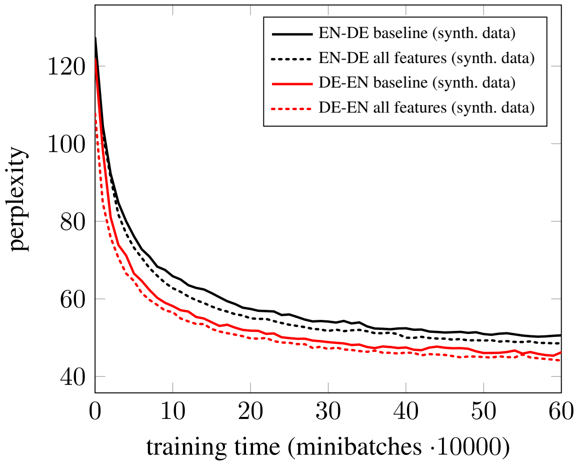


Figure 2: English→German (black) and German→English (red) development set perplexity as a function of training time (number of minibatches) with and without linguistic features.

system	ppl ↓	BLEU ↑	CHRF3 ↑
(Peter et al., 2016)	-	28.9	57.1
baseline	74.9	23.8	52.5
all features	72.7	24.8*	53.5
baseline (+synth. data)	50.9	28.2	56.1
all features (+synth. data)	50.1	29.2*	56.6

Table 5: English→Romanian translation results: best perplexity on newsdev2016, and BLEU and CHRF3 on newstest2016. BLEU scores that are significantly different ($p < 0.05$) from respective baseline are marked with (*).

for different systems (Figure 2). We can see that perplexity is consistently lower for the systems trained with linguistic features.

Table 5 shows results for a lower-resourced language pair, English→Romanian. With linguistic features, we observe improvements of 1.0 BLEU over the baseline, both for the systems trained on parallel data only (23.8→24.8), and the systems which use synthetic training data (28.2→29.2). According to BLEU, the best submission to WMT16 was a system combination by Peter et al. (2016). Our best system is competitive with this submission.

Table 6 shows translation examples of our baseline, and the system augmented with linguistic features. We see that the augmented neural MT systems, in contrast to the respective baselines, successfully resolve the reordering for the German→English example, and the disambiguation of *close* for the English→German example.

system	sentence
source	Gefährlich ist die **Route** aber dennoch.
reference	However **the route** is dangerous.
baseline	Dangerous is **the route**, however.
all features	However, **the route** is dangerous.
source	[We thought] a win like this might be **close**.
reference	[...] dass ein solcher Gewinn **nah** sein könnte.
baseline	[...] ein Sieg wie dieser könnte **schließen**.
all features	[...] ein Sieg wie dieser könnte **nah** sein.

Table 6: Translation examples illustrating the effect of adding linguistic input features.

5 Related Work

Linguistic features have been used in neural language modelling (Alexandrescu and Kirchhoff, 2006), and are also used in other tasks for which neural models have recently been employed, such as syntactic parsing (Chen and Manning, 2014). This paper addresses the question whether linguistic features on the source side are beneficial for neural machine translation. On the target side, linguistic features are harder to obtain for a generation task such as machine translation, since this would require incremental parsing of the hypotheses at test time, and this is possible future work.

Among others, our model incorporates information from a dependency annotation, but is still a sequence-to-sequence model. Eriguchi et al. (2016) propose a tree-to-sequence model whose encoder computes vector representations for each phrase in the source tree. Their focus is on exploiting the (unlabelled) structure of a syntactic annotation, whereas we are focused on the disambiguation power of the functional dependency labels.

Factored translation models are often used in phrase-based SMT (Koehn and Hoang, 2007) as a means to incorporate extra linguistic information. However, neural MT can provide a much more flexible mechanism for adding such information. Because phrase-based models cannot easily generalize to new feature combinations, the individual models either treat each feature combination as an atomic unit, resulting in data sparsity, or assume independence between features, for instance by having separate language models for words and POS tags. In contrast, we exploit the strong generalization ability of neural networks, and expect that even new feature combinations, e.g. a word that appears in a novel syntactic function, are handled gracefully.

One could consider the lemmatized representation of the input as a second source text, and perform multi-source translation (Zoph and Knight, 2016). The main technical difference is that in our approach, the encoder and attention layers are shared between features, which we deem appropriate for the types of features that we tested.

6 Conclusion

In this paper we investigate whether linguistic input features are beneficial to neural machine translation, and our empirical evidence suggests that this is the case.

We describe a generalization of the encoder in the popular attentional encoder-decoder architecture for neural machine translation that allows for the inclusion of an arbitrary number of input features. We empirically test the inclusion of various linguistic features, including lemmas, part-of-speech tags, syntactic dependency labels, and morphological features, into English↔German, and English→Romanian neural MT systems. Our experiments show that the linguistic features yield improvements over our baseline, resulting in improvements on newstest2016 of 1.5 BLEU for German→English, 0.6 BLEU for English→German, and 1.0 BLEU for English→Romanian.

In the future, we expect several developments that will shed more light on the usefulness of linguistic (or other) input features, and whether they will establish themselves as a core component of neural machine translation. On the one hand, the machine learning capability of neural architectures is likely to increase, decreasing the benefit provided by the features we tested. On the other hand, there is potential to explore the inclusion of novel features for neural MT, which might prove to be even more helpful than the ones we investigated, and the features we investigated may prove especially helpful for some translation settings, such as very low-resourced settings and/or translation settings with a highly inflected source language.

Acknowledgments

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreements 645452 (QT21), and 644402 (HimL).

References

Andrei Alexandrescu and Katrin Kirchhoff. 2006. Factored Neural Language Models. In *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*, pages 1–4, New York City, USA. Association for Computational Linguistics.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural Machine Translation by Jointly Learning to Align and Translate. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Danqi Chen and Christopher Manning. 2014. A Fast and Accurate Dependency Parser using Neural Networks. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 740–750, Doha, Qatar. Association for Computational Linguistics.

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar. Association for Computational Linguistics.

Akiko Eriguchi, Kazuma Hashimoto, and Yoshimasa Tsuruoka. 2016. Tree-to-Sequence Attentional Neural Machine Translation. *ArXiv e-prints*.

Çaglar Gülçehre, Orhan Firat, Kelvin Xu, Kyunghyun Cho, Loïc Barrault, Huei-Chi Lin, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2015. On Using Monolingual Corpora in Neural Machine Translation. *CoRR*, abs/1503.03535.

Sébastien Jean, Orhan Firat, Kyunghyun Cho, Roland Memisevic, and Yoshua Bengio. 2015. Montreal Neural Machine Translation Systems for WMT'15 . In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 134–140, Lisbon, Portugal. Association for Computational Linguistics.

Philipp Koehn and Hieu Hoang. 2007. Factored Translation Models. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 868–876, Prague, Czech Republic. Association for Computational Linguistics.

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013. Linguistic Regularities in Continuous Space Word Representations. In *HLT-NAACL*, pages 746–751. The Association for Computational Linguistics.

Guido Minnen, John A. Carroll, and Darren Pearce. 2001. Applied morphological processing of English. *Natural Language Engineering*, 7(3):207–223.

Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. 2013. On the difficulty of training recurrent neural networks. In *Proceedings of the 30th International Conference on Machine Learning, ICML 2013*, pages 1310–1318, Atlanta, USA.

Jan-Thorsten Peter, Tamer Alkhouli, Hermann Ney, Matthias Huck, Fabienne Braune, Alexander Fraser, Aleš Tamchyna, Ondřej Bojar, Barry Haddow, Rico Sennrich, Frédéric Blain, Lucia Specia, Jan Niehues, Alex Waibel, Alexandre Allauzen, Lauriane Aufrant, Franck Burlot, Elena Knyazeva, Thomas Lavergne, François Yvon, and Marcis Pinnis. 2016. The QT21/HimL Combined Machine Translation System. In *Proceedings of the First Conference on Machine Translation (WMT16)*, Berlin, Germany.

Maja Popović. 2015. chrF: character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal. Association for Computational Linguistics.

Stefan Riezler and John T. Maxwell. 2005. On Some Pitfalls in Automatic Evaluation and Significance Testing for MT. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 57–64, Ann Arbor, Michigan. Association for Computational Linguistics.

Helmut Schmid, Arne Fitschen, and Ulrich Heid. 2004. A German Computational Morphology Covering Derivation, Composition, and Inflection. In *Proceedings of the IVth International Conference on Language Resources and Evaluation (LREC 2004)*, pages 1263–1266.

Rico Sennrich and Beat Kunz. 2014. Zmorge: A German Morphological Lexicon Extracted from Wiktionary. In *Proceedings of the 9th International Conference on Language Resources and Evaluation (LREC 2014)*, Reykjavik, Iceland.

Rico Sennrich, Martin Volk, and Gerold Schneider. 2013. Exploiting Synergies Between Open Resources for German Dependency Parsing, POS-tagging, and Morphological Analysis. In *Proceedings of the International Conference Recent Advances in Natural Language Processing 2013*, pages 601–609, Hissar, Bulgaria.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Edinburgh Neural Machine Translation Systems for WMT 16. In *Proceedings of the First Conference on Machine Translation (WMT16)*, Berlin, Germany.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. Improving Neural Machine Translation Models with Monolingual Data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL 2016)*, Berlin, Germany.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016c. Neural Machine Translation of Rare Words with Subword Units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL 2016)*, Berlin, Germany.

Miloš Stanojević, Amir Kamran, Philipp Koehn, and Ondřej Bojar. 2015. Results of the WMT15 Metrics Shared Task. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 256–273, Lisbon, Portugal. Association for Computational Linguistics.

Kristina Toutanova, Dan Klein, Christopher D. Manning, and Yoram Singer. 2003. Feature-Rich Part-of-Speech Tagging with a Cyclic Dependency Network. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*.

Philip Williams, Rico Sennrich, Maria Nadejde, Matthias Huck, Barry Haddow, and Ondřej Bojar. 2016. Edinburgh's Statistical Machine Translation Systems for WMT16. In *Proceedings of the First Conference on Machine Translation (WMT16)*.

Matthew D. Zeiler. 2012. ADADELTA: An Adaptive Learning Rate Method. *CoRR*, abs/1212.5701.

Barret Zoph and Kevin Knight. 2016. Multi-Source Neural Translation. In *NAACL HLT 2016*.