Understanding and Improving Morphological Learning in the Neural Machine Translation Decoder

Fahim Dalvi Nadir Durrani Hassan Sajjad
Yonatan Belinkov* Stephan Vogel

Qatar Computing Research Institute – HBKU, Doha, Qatar
{faimaduddin, ndurrani, hsajjad, svogel}@qf.org.qa

*MIT Computer Science and Artificial Intelligence Laboratory, Cambridge, MA 02139, USA
belinkov@mit.edu

Abstract

End-to-end training makes the neural machine translation (NMT) architecture simpler, yet elegant compared to traditional statistical machine translation (SMT). However, little is known about linguistic patterns of morphology, syntax and semantics learned during the training of NMT systems, and more importantly, which parts of the architecture are responsible for learning each of these phenomena. In this paper we i) analyze how much morphology an NMT decoder learns, and ii) investigate whether injecting target morphology into the decoder helps it produce better translations. To this end we present three methods: i) joint generation, ii) joint-data learning, and iii) multi-task learning. Our results show that explicit morphological information helps the decoder learn target language morphology and improves the translation quality by 0.2–0.6 BLEU points.

1 Introduction

Neural machine translation (NMT) offers an elegant end-to-end architecture, improving translation quality compared to traditional phrase-based machine translation. These improvements are attributed to more fluent output (Toral and Sánchez-Cartagena, 2017) and better handling of morphology and long-range dependencies (Bentivogli et al., 2016). However, systematic studies are required to understand what kinds of linguistic phenomena (morphology, syntax, semantics, etc.) are learned by these models and more importantly, which of the components is responsible for each phenomenon.

A few attempts have been made to understand what NMT models learn about morphology (Belinkov et al., 2017a), syntax (Shi et al., 2016) and semantics (Belinkov et al., 2017b). Shi et al. (2016) used activations at various layers from the NMT encoder to predict syntactic properties on the source-side, while Belinkov et al. (2017a) and Belinkov et al. (2017b) used a similar approach to investigate the quality of word representations on the task of morphological and semantic tagging.

Belinkov et al. (2017a) found that word representations learned from the encoder are rich in morphological information, while representations learned from the decoder are significantly poorer. However, the paper does not present a convincing explanation for this finding. Our first contribution in this work is to provide a more comprehensive analysis of morphological learning on the decoder side. We hypothesize that other components of the NMT architecture – specifically the encoder and the attention mechanism, learn enough information about the target language morphology for the decoder to perform reasonably well, without incorporating high levels of morphological knowledge into the decoder. To probe this hypothesis, we investigate the following questions:

- What is the effect of attention on the performance of the decoder?
- How much does the encoder help the decoder in predicting the correct morphological variant of the word it generates?

To answer these questions, we train NMT models for different language pairs, involving morphologically rich languages such as German and Czech. We then use the trained models to extract features from the decoder for words in the language of interest. Finally we train a classifier using the extracted features to predict the morphological tag of the words. The accuracy of this ex-

ternal classifier gives us a quantitative measure of how well the NMT model learned features that are relevant to morphology. Our results indicate that both the encoder and the attention mechanism aid the decoder in generating correct morphological forms, and thus limit the need of the decoder to learn target morphology.

Motivated by these findings, we hypothesize that it may be possible to force the decoder to learn more about morphology by injecting the morphological information during training which can in turn improve the overall translation quality. In order to test this hypothesis, we experiment with three possible solutions:

1. *Joint Generation*: An NMT model is trained on the concatenation of words and morphological tags on the target side.

2. *Joint-data learning*: An NMT model is trained where each source sequence is used twice with an artificial token to either predict target words or morphological tags.

3. *Multi-task learning*: A multi-task NMT system with two objective functions is trained to jointly learn translation and morphological tagging.

Our experiments show that word representations learned after explicitly injecting target morphology improve morphological tagging accuracy of the decoder by 3% and also improves the translation quality by up to 0.6 BLEU points.

The remainder of this paper is organized as follows. Section 2 describes our experimental setup. Section 3 shows an analysis of the decoder. Section 4 describes the three proposed methods to integrate morphology into the decoder. Section 5 presents the results. Section 6 gives an account of related work and Section 7 concludes the paper.

2 Experimental Design

Parallel Data

We used the German-English and Czech-English datasets from the WIT[3] TED corpus (Cettolo, 2016) made available for IWSLT 2016. We used the official training sets to analyze and evaluate the proposed methods for integrating morphology . The corpus also provides four test sets, test-11 through test-14. We used test-11 for tuning, and the other test sets for evaluation. The statistics for the sets are provided in Table 1.

Language-pair	Sentences	tok$_{de/cz}$	tok$_{en}$
De↔En	210K	4M	4.2M
Cz↔En	122K	2.1M	2.5M

Table 1: Statistics for the data used for training, tuning and testing

Morphological Annotations

In order to train and evaluate the external classifier on the extracted features, we required data annotated with morphological tags. We used the following tools recommended on the Moses website[1] to annotate the data: LoPar (Schmid, 2000) for German, Tree-tagger (Schmid, 1994) for Czech and MXPOST (Ratnaparkhi, 1998) for English. The number of tags produced by these taggers is 214 for German and 368 for Czech.

Data preprocessing

We used the standard MT pre-processing pipeline of tokenizing and truecasing the data using Moses (Koehn et al., 2007) scripts. We did not apply byte-pair encoding (BPE) (Sennrich et al., 2016b), which has recently become a common part of the NMT pipeline, because both our analysis and the annotation tools are word level.[2] However, experimenting with BPE and other representations such as character-based models (Kim et al., 2015) would be interesting.[3]

NMT Systems

We used the `seq2seq-attn` implementation (Kim, 2016) with the following default settings: word embeddings and LSTM states with 500 dimensions, SGD with an initial learning rate of 1.0 and decay rate of 0.5 (after the 9th epoch), and dropout rate of 0.3. We use two uni-directional

[1]These have been used frequently to annotate data in the previous evaluation campaigns (Birch et al., 2014; Durrani et al., 2014).

[2]The difficulty with using these is that it is not straightforward to derive word representations out of a decoder that processes BPE-ed text, because the original words are split into subwords. We considered aggregating the representations of BPE subword units, but the choice of aggregation strategy may have an undesired impact on the analysis. For this reason we decided to leave exploration of BPE for future work.

[3]Character-based models are becoming increasingly popular in Neural MT, for addressing the rare word problem – and they have been used previously also to benefit MT for morphologically rich (Luong et al., 2010; Belinkov and Glass, 2016; Costa-jussà and Fonollosa, 2016) and closely related languages (Durrani and Koehn, 2014; Sajjad et al., 2013).

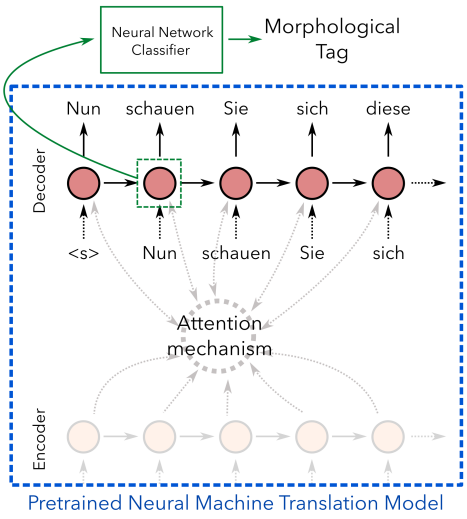

Figure 1: Features for the word *Nun* ($\texttt{DEC}_{t_1}$) are extracted from the decoder of a pre-trained NMT system and provided to the classifier for training

hidden layers for both the encoder and the decoder. The NMT system is trained for 13 epochs, and the model with the best validation loss is used for extracting features for the external classifier. We use a vocabulary size of 50000 on both the source and target side.

Classifier Settings

For the classification task, we used a feed-forward network with one hidden layer, dropout ($\rho = 0.5$), a ReLU non-linearity, and an output layer mapping to the tag set (followed by a Softmax). The size of the hidden layer is set to be identical to the size of the NMT decoder's hidden state (500 dimensions). The classifier has no explicit access to context other than the hidden representation generated by the NMT system, which allows us to focus on the quality of the representation. We use Adam (Kingma and Ba, 2014) with default parameters to minimize the cross-entropy objective.

3 Decoder Analysis

3.1 Methodology

We follow a process similar to Shi et al. (2016) and Belinkov et al. (2017a) to analyze the NMT systems but with a focus on the decoder component of the architecture. Formally, given a source sentence $s = \{s_1, s_2, ..., s_N\}$ and a target sentence $t = \{t_1, t_2, ..., t_M\}$, we first use the encoder (Equation 1) to compute a set of hidden states $h = \{h_1, h_2, ..., h_N\}$. We then use an attention mechanism (Bahdanau et al., 2014) to compute a weighted average of these hidden states from the previous decoder state (d_{i-1}), known as the context vector c_i (Equation 2). The context vector is a real valued vector of k dimensions, which is set to be the same as the hidden states in our case. The attention model computes a weight w_{h_i} for each hidden state of the encoder, thus giving soft alignment for each target word. The context vector is then used by the decoder (Equation 3) to generate the next word in the target sequence:

$$\texttt{ENC} : s = \{s_1, ..., s_N\} \mapsto h = \{h_1, ..., h_N\} \quad (1)$$

$$\texttt{ATTN}_i : h, d_{i-1}, t_{i-1} \mapsto c_i \in \mathbb{R}^k (1 \leq i \leq M) \quad (2)$$

$$\texttt{DEC} : \{c_1, ..., c_M\} \mapsto t = \{t_1, t_2, ..., t_M\} \quad (3)$$

After training the NMT system, we freeze the parameters of the network and use the encoder or the decoder as a feature extractor to generate vectors representing words in the sentence. Let $\texttt{ENC}_{s_i}$ denote the representation of a source word s_i. We use $\texttt{ENC}_{s_i}$ to train the external classifier that for predicting the morphological tag for s_i and evaluate the quality of the representation based on our ability to train a good classifier. For word representations on the target side, we feed our word of interest t_i as the previously predicted word, and extract the representation $\texttt{DEC}_{t_i}$ from the higher layers (See Figure 1 for illustration).

Note that in the decoder, the target word representations $\texttt{DEC}_{t_i}$ are not learned for predicting the word t_i, but the next word (t_{i+1}). Hence, it is arguable that $\texttt{DEC}_{t_i}$ actually captures morphological information about t_{i+1} rather than t_i, which can also explain the poorer decoder accuracies. To test this argument, we also trained our systems assuming that $\texttt{DEC}_{t_i}$ encodes morphological information about the next word t_{i+1}. In this case, the decoder performance dropped by almost 15%. $\texttt{DEC}_{t_i}$ probably encodes morphological information about both the current word (t_i) and the next word (t_{i+1}). However, we leave this exploration for future work, and work with the assumption that $\texttt{DEC}_{t_i}$ encodes information about word t_i.

3.2 Analysis

Before diving into the decoder's performance, we first compare the performance of encoder versus decoder by training De↔En[4] and Cz↔En

[4]By De↔En, we mean independently trained German-to-English and English-to-German models.

Baseline	$\texttt{ENC}_{s_i}$	$\texttt{DEC}_{t_i}$
De↔En	89.5	44.55
Cz↔En	77.0	36.35

Table 2: Comparison of morphological accuracy for the encoder and decoder representations

	$\texttt{DEC}_{t_i}$	w/o-ATTN	$\texttt{DEC}_{t_i}$+$\texttt{ENC}_{t_i}$	$\texttt{ENC}_{t_i}$
En→De	44.55	50.26	60.34	43.43
En→Cz	36.35	42.09	48.64	36.36

Table 3: Morphological Tagging accuracy of the Decoder with and without attention, and effect of considering the most attended source word ($\texttt{ENC}_{t_i}$)

NMT models. We use the De→En/Cz→En models to extract encoder representations, and the En→De/En→Cz models to extract decoder representations. We then feed these representations to our classifier to predict morphological tags for German and Czech words. Table 2 shows that German and Czech representations learned on the encoder-side (using the De→En/Cz→En models) give much better accuracy compared to the ones learned on the decoder-side (using the En→De/En→Cz models).

Given this difference in performance between the two components in our NMT system, we analyze the decoder further in various settings: comparing the performance i) with and without the attention mechanism, and ii) augmenting the decoder representation with the representation of the most attended source word. The baseline NMT models were trained with an attention mechanism. In an attempt to probe what effect the attention mechanism has on the decoder's performance in the context of learning target language morphology, we trained NMT models without attention. Next we tried to take our baseline model (with attention) and augment its decoder representations with the encoder hidden state corresponding to the maximum attention (hereby denoted as $\texttt{ENC}_{t_i}$). Our hypothesis is that since the decoder focuses on this hidden state to output the next target word, it may also encode some useful information about target morphology. Lastly, we also train a classifier on $\texttt{ENC}_{t_i}$ alone in order to compare the ability of the encoder and decoder in learning *target* language morphology.

Table 3 summarizes the results of these experiments. Comparing systems with ($\texttt{DEC}_{t_i}$) and without attention (w/o-ATTN), we see that the accuracy on the morphological tagging task goes up when no attention is used. This can be explained by the fact that in the case of no attention, the decoder only receives a single context vector from the encoder and it has to learn more information about each target word to make accurate predictions. It is difficult for the encoder to transfer information about each target word using the same context vector cleanly, causing the decoder to learn more, resulting in better decoder performance in regards to the morphological information learned.

The second part of the table presents results involving encoder representations to aid morphological analysis of target words. There is a significant boost in the classifier's performance when the decoder representation for a target word t_i is concatenated with the encoder representation of the most attended source word ($\texttt{DEC}_{t_i}$+$\texttt{ENC}_{t_i}$). This hints towards several hypotheses: i) because the source and target words are translations, they share some morphological properties (e.g. nouns get translated to nouns, etc.), ii) the encoder also learns and stores information about the target language, so that the attention mechanism can make use of this information while deciding which word to focus on next. To ensure that the encoder and decoder indeed learn different information, we also tried to classify the morphological tag of a given word t_i based on the encoder representation of the most attended source word alone ($\texttt{ENC}_{t_i}$). We see a drop in accuracy, showing that both encoder and decoder learned different things about the same target word and are complementary representations. We can also see that the accuracy of the combined representation ($\texttt{DEC}_{t_i}$+$\texttt{ENC}_{t_i}$) still lags behind the encoder's performance in predicting source morphology (Table 2). This indicates that there is still room for improvement in the NMT model's ability to learn target side morphology.

In this section, we showed that the encoder and decoder learn different amounts of morphology due to the varying nature of their tasks within NMT architecture. The decoder depends on the encoder and attention mechanism to generate the correct morphological variant of a target word.

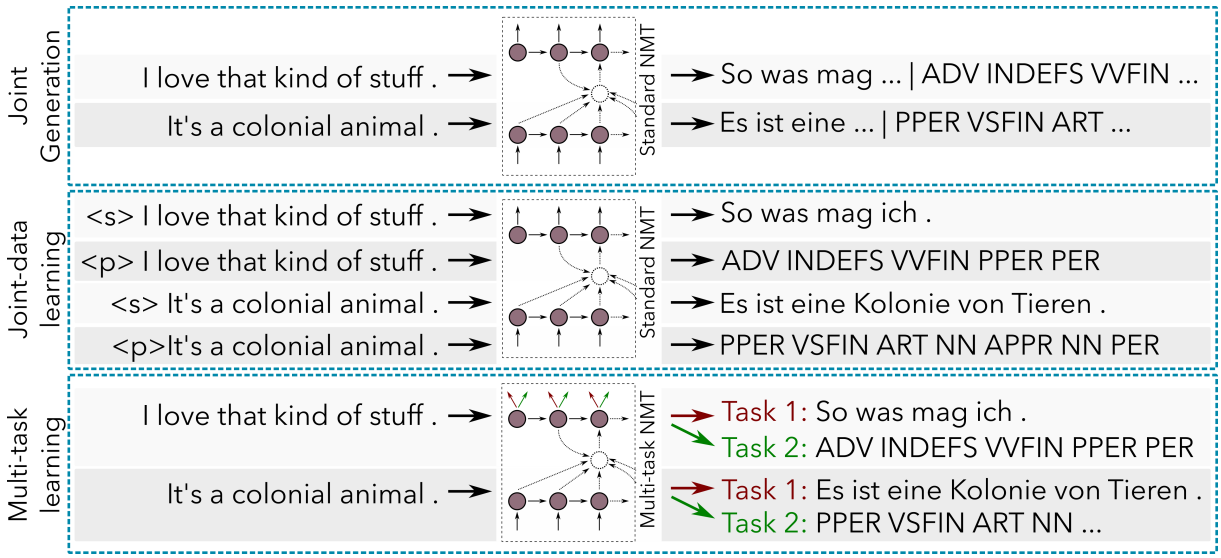

Figure 2: Various approaches to inject morphological knowledge into the decoder

4 Morphology-aware Decoder

Motivated by the result that the decoder learns considerably less amount of morphology than the encoder (Table 2) and the overall system does not learn as much about target morphology as source morphology, we investigated three ways to directly inject target morphology into the decoder, namely: i) Joint Generation, ii) Joint-data Learning, iii) Multi-task Learning. Figure 2 illustrates the approaches.

4.1 Joint Generation

As our first approach, we considered a solution that uses the standard NMT architecture, but is trained on a modified dataset. To incorporate morphological information, we modify the target sentence by appending the morphological tag sequence to it. The NMT system trained on this data learns to produce both words and morphological tags simultaneously. Formally, given a source sentence $s = \{s_1, ..., s_N\}$, target sentence $t = \{t_1, ..., t_M\}$ and its morphological sequence $m = \{m_1, ..., m_M\}$, we train an NMT system on (s', t') pairs, where $s' = s$ and $t' = t + m$. Although this model is quite weak and the (word and morphological) bases are quite far away, we posit that the attention mechanism might be able to attend to the same source word twice. Given this, the decoder gets a similar representation from which it has to predict a word in the first instance, and a tag in the second - thus helping in common learning for the two tasks.

4.2 Joint-data Learning

Given the drawbacks of the first approach, we considered another data augmentation technique inspired by multilingual NMT systems (Johnson et al., 2016). Instead of having multiple source and target languages, we used one source language and two target language variations. The training data consists of sequences of source→target words and source→target morphological tags. We added an artificial token in the beginning of each source sentence indicating whether we want to generate target words or morphological tags. Using an artificial token in the source sentence has been explored and shown to work well to control the style of the target language (Sennrich et al., 2016a). The objective function is the same as the one in usual sequence-to-sequence models, and is hence shared to minimize both morphological and translation error given the mixed data.

4.3 Multi-task Learning

In this final method, we decided to follow a more principled approach and modified the standard sequence-to-sequence for multi-task learning. The goal in multi-task training is to learn several tasks simultaneously such that each task can benefit from the mutual information learned (Collobert and Weston, 2008). [5] With this motivation, we modified the NMT decoder to predict not only a word but also its corresponding tag. All of the layers below the output layers are shared. We have

[5] For example, Eriguchi et al. (2017) jointly learned the tasks of parsing and translation.

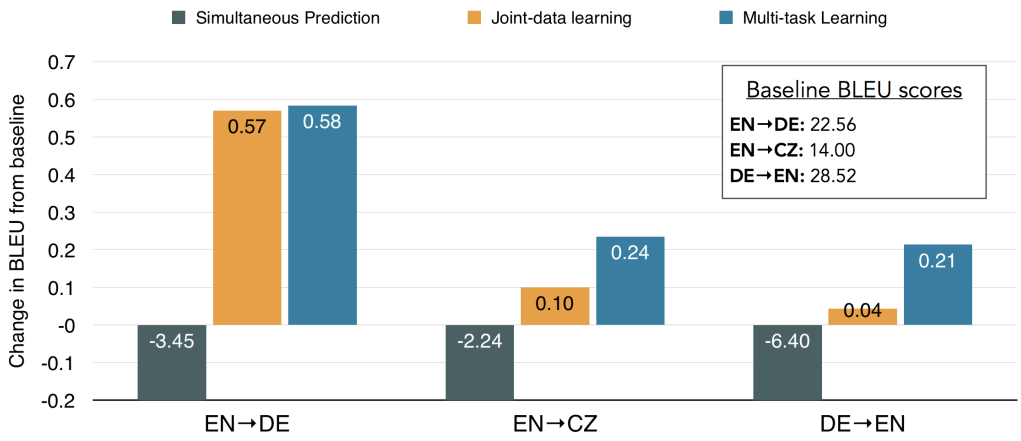

Figure 3: Improvements from adding morphology. A y-value of zero represents the baseline

two output layers in parallel – the first to predict the target word, and the second to predict the morphological tag of the target word. Both ouput layers have their own separate loss function. While training, we combine the losses from both output layers to jointly train the system. This is different from the Joint-data learning technique, where we predict entire sequences of words or tags without any dependence on each other.

Formally, given a set of N tasks, sequence-to-sequence multi-task learning involves an objective function minimizing the overall loss, which is a weighted combination of the N individual task losses. In our scenario, the training corpus consisted of a multi-target corpus: source→target words and source→target morphological tags, i.e $N = 2$. Hence, given a set of training examples $D = \{\langle s^{(n)}, t^{(n)}, m^{(n)}\rangle\}_{n=1}^{N}$, where s is the source sentence, t is the target sentence and m is the target morphological tag sequence, the new objective function to maximize is as follows:

$$\mathcal{L} = (1-\lambda)\sum_{n=1}^{N}\log P(t^{(n)}|s^{(n)};\theta) + \lambda\sum_{n=1}^{N}\log P(m^{(n)}|s^{(n)};\theta)$$

Where λ is a hyper-parameter used to shift focus towards translation or the morphological tagging.[6]

5 Results and Discussion

Our results show that the multi-task learning approach performed the best among the three approaches, while the Joint Generation method has the poorest performance. Figure 3 summarizes the results for different language pairs. The joint generation method degrades overall translation performance, as expected, given its weakness from a modeling perspective. It is possible that even though the attention mechanism is able to focus on the source sequence in two passes, the parts of the network that predict words and tags are not tightly coupled enough to learn from each other.

The BLEU scores improved when using the other two methods. We achieved an improvement of up to 0.6 BLEU points and 3% (in tagging accuracy). The best improvements were obtained in the En→De direction, while we observed lesser gains in the De→En. This is perhaps because English is morphologically poorer, and the baseline system was able to learn the required amount of morphological information from the text itself. Improvements were also obtained for the En→Cz direction, although not as much as in German. This could be due to data sparsity: Czech is much richer in morphology,[7] and the available TED En↔Cz data was 40% less than the En↔De data.

Joint-data vs. Multi-task Learning

Both Joint-data learning and Multi-task learning improved overall translation performance. In the case of En→De, the performance of both approaches is very similar. However, each has its own pros and cons. While the joint-data learning method is a simple approach that allows to add morphology and other linguistic information without needing to change the architecture, the multi-task learning approach is a more principled and

[6] We tuned the weight parameter on held-out data.

[7] The number of morphological tags in Czech are 368 versus 214 in German.

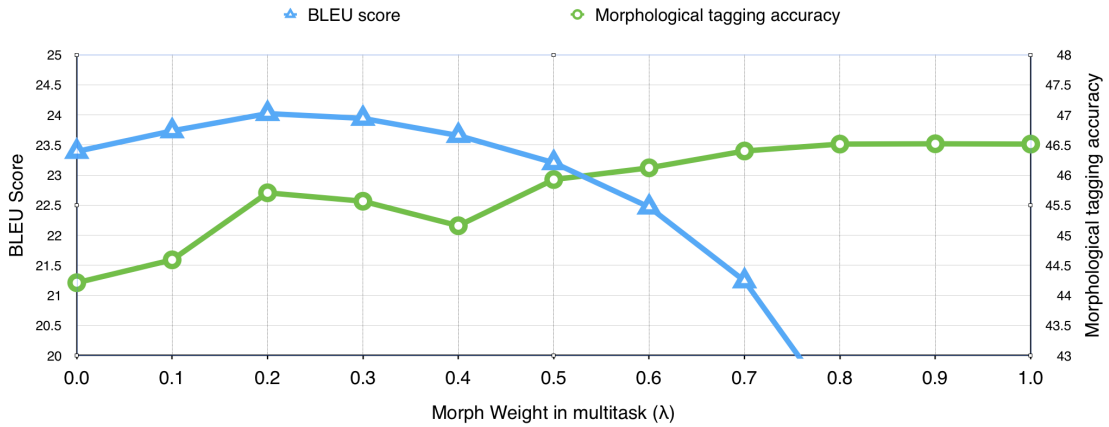

Figure 4: Multi-task learning: Translation vs. Morphological Tagging weight for En→De model

powerful way of integrating the same information into the decoder. Having separate objective functions in multi-task learning also allows us to adjust the balance between the two tasks, which can be handy if the morphological information quality is not very high. On the flip side, this additional explicit weight adjustment can also be viewed as a potential constraint that is not present in the joint-data learning approach.

Multi-task Weight Hyper-Parameter

As discussed, the multi-task learning approach has an additional weight hyper-parameter λ that adjusts the balance between word and tag prediction. Figure 4 shows the result of varying λ from no morphological information ($\lambda = 0$) to only morphological information ($\lambda = 1$) on test-11 set. The left y-axis presents the BLEU score and the right y-axis presents the morphological accuracy. The best morphological accuracy is achieved at $\lambda = 1$ which does not correspond to best translation quality since at that point the model is only minimizing the tag objective function. Similarly at $\lambda = 0$, the model falls back to the baseline model with a single objective function minimizing translation error. For all language pairs, we consistently achieved the best BLEU score at $\lambda = 0.2$. The parameter was tuned on a separate held out development set (test-11), and the results shown in Figure 3 are on blind test sets (test-12,13). Averages are reported in the figure.

6 Related Work

The related work to this paper can be broken into two groups:

Analysis Several approaches have been devised to analyze MT models and the linguistic properties that are learned during training. A common approach has been to use activations from a trained model to train an external classifier to predict some relevant information about the input. Köhn (2015) and Qian et al. (2016b) analyzed linguistic information learned in word embeddings, while Qian et al. (2016a) went further and analyzed linguistic properties in the hidden states of a recurrent neural network. Adi et al. (2016) looked at the overall information learned in a sentence summary vector generated by an RNN using a similar approach. Our approach closely aligns with that of Shi et al. (2016) and Belinkov et al. (2017a), where the activations from various layers in a trained NMT system are used to predict linguistic properties.

Integrating Morphology Some work has also been done in injecting morphological or more general linguistic knowledge into an NMT system. Sennrich and Haddow (2016) proposed a factored model that incorporates linguistic features on the source side as additional factors. An embedding is learned for each factor, just like a source word, and then the word and factor embeddings are combined before being passed on to the encoder. Aharoni and Goldberg (2017) proposed a method to predict the target sentence along with its syntactic tree. They linearize the tree in order to use the existing sequence-to-sequence model. Nadejde et al. (2017) also evaluated several methods of incorporating syntactic knowledge on both the source and target. While they used factors on the source side, their best method for the target side was to linearize the information and interleave it between the target words. García-Martínez et al.

(2016) used a neural MT model with multiple outputs, like in our case of *Multi-task learning*. Their model predicts two properties at every step, the lemma of the target word and its morphological information. They then use an external tool to use this information to generate the actual target word. Dong et al. (2015) presented multi-task learning to translate a language into multiple target languages, and Luong et al. (2015) did experiments involving several levels of source and target language information. There have been previous efforts to integrate morphology into MT systems by learning factored models (Koehn and Hoang, 2007; Durrani et al., 2015) over POS and morphological tags.

7 Conclusion

In this paper we analyzed and investigated ways to improve morphological learning in the NMT decoder. We carried a series of experiments to understand why the decoder learns considerably less morphology than the encoder in the NMT architecture. We found that the decoder needs assistance from the encoder and the attention mechanism to generate correct target morphology. Additionally we explored three ways to explicitly inject morphology in the decoder: joint generation, joint-data learning, and multi-task learning. We found multi-task learning to outperform the other two methods. The simpler joint-data learning method also gave decent improvements. The code for the experiments and the modified framework is available at `https://github.com/fdalvi/seq2seq-attn-multitask`.

References

Yossi Adi, Einat Kermany, Yonatan Belinkov, Ofer Lavi, and Yoav Goldberg. 2016. Fine-grained Analysis of Sentence Embeddings Using Auxiliary Prediction Tasks. *arXiv preprint arXiv:1608.04207*.

Roee Aharoni and Yoav Goldberg. 2017. Towards String-To-Tree Neural Machine Translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 132–140. Association for Computational Linguistics.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural Machine Translation by Jointly Learning to Align and Translate. *arXiv preprint arXiv:1409.0473*.

Yonatan Belinkov, Nadir Durrani, Fahim Dalvi, Hassan Sajjad, and James Glass. 2017a. What do Neural Machine Translation Models Learn about Morphology? In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 861–872. Association for Computational Linguistics.

Yonatan Belinkov and James Glass. 2016. Large-Scale Machine Translation between Arabic and Hebrew: Available Corpora and Initial Results. In *Proceedings of the Workshop on Semitic Machine Translation*, pages 7–12, Austin, Texas. Association for Computational Linguistics.

Yonatan Belinkov, Lluis Marquez, Hassan Sajjad, Nadir Durrani, Fahim Dalvi, and James Glass. 2017b. Evaluating layers of representation in neural machine translation on parts-of-speech and semantic tagging task. In *Proceedings of the 8th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, Taipei, Taiwan. Association for Computational Linguistics.

Luisa Bentivogli, Arianna Bisazza, Mauro Cettolo, and Marcello Federico. 2016. Neural versus Phrase-Based Machine Translation Quality: a Case Study. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 257–267, Austin, Texas. Association for Computational Linguistics.

Alexandra Birch, Matthias Huck, Nadir Durrani, Nikolay Bogoychev, and Philipp Koehn. 2014. Edinburgh SLT and MT system description for the IWSLT 2014 evaluation. In *Proceedings of the 11th International Workshop on Spoken Language Translation*, IWSLT '14, Lake Tahoe, CA, USA.

Mauro Cettolo. 2016. An Arabic-Hebrew parallel corpus of TED talks. In *Proceedings of the AMTA Workshop on Semitic Machine Translation (SeMaT)*, Austin, US-TX.

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*, ICML '08, pages 160–167, New York, NY, USA. ACM.

Marta R. Costa-jussà and José A. R. Fonollosa. 2016. Character-based Neural Machine Translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 357–361, Berlin, Germany. Association for Computational Linguistics.

Daxiang Dong, Hua Wu, Wei He, Dianhai Yu, and Haifeng Wang. 2015. Multi-Task Learning for Multiple Language Translation. In *ACL (1)*.

Nadir Durrani, Barry Haddow, Philipp Koehn, and Kenneth Heafield. 2014. Edinburgh's phrase-based machine translation systems for WMT-14. In *Proceedings of the ACL 2014 Ninth Workshop on Statistical Machine Translation*, pages 97–104, Baltimore, MD, USA.

Nadir Durrani and Philipp Koehn. 2014. Improving Machine Translation via Triangulation and Transliteration. In *Proceedings of the 17th Annual Conference of the European Association for Machine Translation*, EAMT'14, pages 71–78, Dubrovnik, Croatia.

Nadir Durrani, Helmut Schmid, Alexander Fraser, Philipp Koehn, and Hinrich Schütze. 2015. The Operation Sequence Model – Combining N-Gram-based and Phrase-based Statistical Machine Translation. *Computational Linguistics*, 41(2):157–186.

Akiko Eriguchi, Yoshimasa Tsuruoka, and Kyunghyun Cho. 2017. Learning to parse and translate improves neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 72–78. Association for Computational Linguistics.

Mercedes García-Martínez, Loïc Barrault, and Fethi Bougares. 2016. Factored neural machine translation. *CoRR*, abs/1609.04621.

Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda B. Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's multilingual neural machine translation system: Enabling zero-shot translation. *CoRR*, abs/1611.04558.

Yoon Kim. 2016. Seq2seq-attn. `https://github.com/harvardnlp/seq2seq-attn`.

Yoon Kim, Yacine Jernite, David Sontag, and Alexander M Rush. 2015. Character-aware Neural Language Models. *arXiv preprint arXiv:1508.06615*.

Diederik Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. *arXiv preprint arXiv:1412.6980*.

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.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the Association for Computational Linguistics (ACL'07)*, Prague, Czech Republic.

Arne Köhn. 2015. What's in an Embedding? Analyzing Word Embeddings through Multilingual Evaluation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2067–2073, Lisbon, Portugal. Association for Computational Linguistics.

Minh-Thang Luong, Quoc V. Le, Ilya Sutskever, Oriol Vinyals, and Lukasz Kaiser. 2015. Multi-task sequence to sequence learning. *CoRR*, abs/1511.06114.

Minh-Thang Luong, Preslav Nakov, and Min-Yen Kan. 2010. A Hybrid Morpheme-Word Representation for Machine Translation of Morphologically Rich Languages. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 148–157. Association for Computational Linguistics.

Maria Nadejde, Siva Reddy, Rico Sennrich, Tomasz Dwojak, Marcin Junczys-Dowmunt, Philipp Koehn, and Alexandra Birch. 2017. Syntax-aware neural machine translation using CCG. *CoRR*, abs/1702.01147.

Peng Qian, Xipeng Qiu, and Xuanjing Huang. 2016a. Analyzing Linguistic Knowledge in Sequential Model of Sentence. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 826–835, Austin, Texas. Association for Computational Linguistics.

Peng Qian, Xipeng Qiu, and Xuanjing Huang. 2016b. Investigating Language Universal and Specific Properties in Word Embeddings. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1478–1488, Berlin, Germany. Association for Computational Linguistics.

Adwait Ratnaparkhi. 1998. *Maximum Entropy Models for Natural Language Ambiguity Resolution*. Ph.D. thesis, University of Pennsylvania, Philadelphia, PA.

Hassan Sajjad, Kareem Darwish, and Yonatan Belinkov. 2013. Translating Dialectal Arabic to English. In *Proceedings of the 51st Conference of the Association for Computational Linguistics (ACL)*.

Helmut Schmid. 1994. Part-of-Speech Tagging with Neural Networks. In *Proceedings of the 15th International Conference on Computational Linguistics (Coling 1994)*, pages 172–176, Kyoto, Japan. Coling 1994 Organizing Committee.

Helmut Schmid. 2000. LoPar: Design and Implementation. Bericht des Sonderforschungsbereiches "Sprachtheoretische Grundlagen fr die Computerlinguistik" 149, Institute for Computational Linguistics, University of Stuttgart.

Rico Sennrich and Barry Haddow. 2016. Linguistic input features improve neural machine translation. In *Proceedings of the First Conference on Machine Translation*, pages 83–91, Berlin, Germany. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Controlling Politeness in Neural Machine Translation via Side Constraints. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics:*

Human Language Technologies, San Diego, California.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. Neural Machine Translation of Rare Words with Subword Units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Xing Shi, Inkit Padhi, and Kevin Knight. 2016. Does String-Based Neural MT Learn Source Syntax? In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1526–1534, Austin, Texas. Association for Computational Linguistics.

Antonio Toral and Víctor M. Sánchez-Cartagena. 2017. A Multifaceted Evaluation of Neural versus Phrase-Based Machine Translation for 9 Language Directions. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 1063–1073, Valencia, Spain. Association for Computational Linguistics.