sciendo

PBML

The Prague Bulletin of Mathematical Linguistics
NUMBER 113 OCTOBER 2019 31–40

Replacing Linguists with Dummies: A Serious Need for Trivial Baselines in Multi-Task Neural Machine Translation

Daniel Kondratyuk, Ronald Cardenas, Ondřej Bojar

Charles University, Faculty of Mathematics and Physics, Institute of Formal and Applied Linguistics

Abstract

Recent developments in machine translation experiment with the idea that a model can improve the translation quality by performing multiple tasks, e.g., translating from source to target and also labeling each source word with syntactic information. The intuition is that the network would generalize knowledge over the multiple tasks, improving the translation performance, especially in low resource conditions. We devised an experiment that casts doubt on this intuition. We perform similar experiments in both multi-decoder and interleaving setups that label each target word either with a syntactic tag or a completely random tag. Surprisingly, we show that the model performs nearly as well on uncorrelated random tags as on true syntactic tags. We hint some possible explanations of this behavior.

The main message from our article is that experimental results with deep neural networks should always be complemented with trivial baselines to document that the observed gain is not due to some unrelated properties of the system or training effects. True confidence in where the gains come from will probably remain problematic anyway.

1. Introduction

Neural models (NMT) have become the default choice for Machine Translation for language pairs with enough parallel data. Even when linguistic phenomena are not explicitly modeled, sequence-to-sequence models appear to implicitly learn some notions of syntax, word order and morphology (Bentivogli et al., 2016; Shi et al., 2016). Recent work explores strategies of incorporating linguistic structure by accounting for it in the architecture itself or by jointly learning auxiliary tasks.

 Corresponding author: bojar@ufal.mff.cuni.cz
Cite as: Daniel Kondratyuk, Ronald Cardenas, Ondřej Bojar. Replacing Linguists with Dummies: A Serious Need for Trivial Baselines in Multi-Task Neural Machine Translation. The Prague Bulletin of Mathematical Linguistics No. 113, 2019, pp. 31–40. doi: 10.2478/pralin-2019-0005.

Obama	receives	Netanyahu	in	the	capital	of	USA
NP	((S[dcl] \NP)/PP)/NP	NP	PP/NP	NP\N	N	(NP\NP) / NP	NP

Figure 1. Example of CCG supertags for English, taken from (Nadejde et al., 2017)

On one hand, previous work proposes to replace the input source token sequence by its parse tree representation, namely RNN Grammar (Dyer et al., 2016), having the source language parser be pre-trained beforehand (Bradbury and Socher, 2017; Eriguchi et al., 2016) or jointly trained with the MT task (Eriguchi et al., 2017). Moreover, there have been some efforts to include syntactic structure priors for better machine translation. Bradbury and Socher (2017) include reinforcement learning to induce unsupervised tree structures on both the source and target sentences. Eriguchi et al. (2016) replace the encoder of an attentional NMT architecture with variants of the TreeLSTM (Tai et al., 2015) in order to account for phrase structure. The results, however, are mixed and mostly evaluated on small parallel corpora.

On the other hand, another line of research explores the contribution of learning simpler downstream tasks in addition to NMT on the source or target side. In such a multi-task scenario,[1] Niehues and Cho (2017) explore the behavior of multitasking on the target side with increasing degrees of sharing of task specific modules (e.g. attention mechanisms, decoders). With a similar goal, Nadejde et al. (2017) proposed a way of tightly coupling syntactic information with token words. They present an *interleaved* setup in which each English token (or BPE segmentation) is preceded by its CCG supertag (Combinatory Categorial Grammar; Steedman, 2000). They report encouraging results when using English on the source or target side.

In this paper, we explore the behavior of sequence-to-sequence architectures with recurrent neural networks (Bahdanau et al., 2014) and Transformer (Vaswani et al., 2017) as underlying blocks in interleaved and multitasking setups. The task to be interleaved or jointly learned is tagging of CCG supertags in the target side for the German-English language pair. We compare scenarios in which the gold tag sequences are actual CCG tags, random tags, and a single repeated tag. In this way, we seek to find out if the model is indeed learning syntactic phenomena that contribute to the translation task. However, we report that, counter-intuitively, jointly learning random tags yields comparable, if not better, results in all the setups explored.

2. Multi-Task Neural Machine Translation

We consider the multi-task approach of jointly learning to translate and tag the target with CCG supertags. Combinatory Categorial Grammar, introduced by Steedman (2000), is a lexicalised formalism that encodes sentence-level morpho-syntactic

[1]Not to be mistaken with *multi-lingual* MT which tackles the problem of translating into or from several languages at the same time.

information in every tag, referred to as *supertag*. Figure 1 shows how the formalism captures information about surrounding syntactic subtree's nodes in the tag itself.

We explore two architecture configurations and two task coupling strategies. The first model architecture we consider is the standard Seq2Seq model with attention. The second one is the Transformer model. Following the setup proposed by Nadejde et al. (2017), only word tokens are split using *byte-pair-encoding* (BPE) (Sennrich et al., 2016) , i.e. CCG tags remain unsplit.

2.1. Muti-Decoder Model

For the first set of experiments, we adopt the multi-decoder model seen in Niehues and Cho (2017) and Nadejde et al. (2017), where the encoder is shared between the two tasks. We then split the network into two decoders, each with their own attention layer on the encoded words. The first decoder predicts the target translation, sharing the word embeddings of the encoder. The second decoder predicts the target language tags using a separate tag vocabulary. Since only word tokens are split using BPE codes and not CCG tags, both decoders may predict sequences of different length. The total loss is then the sum of the two losses from both decoders.

2.2. Interleaved Model

In the second set of experiments, we adopt the interleaved setup proposed by Nadejde et al. (2017). We start with a standard encoder-decoder architecture, and only modify the dataset. We insert a target language tag preceding each sequence of BPE tokens corresponding to a single word token. We then combine the two vocabularies. This requires the network to predict each tag as an additional word to be included in the translation. We also ensure that the tag vocabulary does not overlap with the target language vocabulary in the embedding table.

3. Experiments

We use the Neural Monkey framework (Helcl and Libovický, 2017) for all our experiments. We extend the framework to meet our needs regarding the interleaved setups (see Section 3.4). Translation performance is measured in terms of BLEU (Papineni et al., 2002) as calculated by *multi-bleu.perl*.

3.1. Dataset

We use the English-German parallel corpus of the WMT 2016, tokenized with Moses tokenizer (Koehn et al., 2007). For development, we use the 2013 test set, *news-test2013*. For testing, we use the official 2016 test set, *news-test2016*.

The CCG tagging was done using EasySRL (Lewis et al., 2015) and its pre-trained models for English, setting a sentence length threshold of 74 tokens. Sentences that

could not be parsed were discarded.[2] As sanity check, we test the CCG supertag tagging performance of the EasySRL parser on section 23 of the CCGbank (Hockenmaier and Steedman, 2007). We obtain an accuracy of 70.83% and an F1 score of 73.28%.

The CCG tag vocabulary was limited to 500 tags, the rest being tagged as UNK. The UNK token appears 99 times in the training data (out of 100M total tag tokens). The final number of parallel sentences was 4,473,920 in the training set, 2,986 in the development set, and 2,994 in the test set.

3.2. Tag Schemes

Additionally, we experiment with three types of tag schemes for the target English dataset. The first tag scheme uses the CCG supertags of each target word for prediction.

The second tag scheme uses random tags, keeping the vocabulary size the same. We generate random tag ids in the range [0-499] by sampling from the uniform distribution without replacement within each sentence. To see the effect of randomness on our models, we define a third tag scheme which effectively maps all tags in the dataset to a single token, i.e., tag id 0. We fix the vocabulary size to 500 (and not size 1) so that the results are comparable.

3.3. Baselines and Setups

We consider the single-task NMT architectures as baselines: Seq2Seq and Transformer. We set up our experiments by varying the following three aspects of the pipeline:

- **Architecture**: Seq2Seq, Transformer
- **Multi-Task Configuration**: multi-decoder, interleaved.
- **Tag Scheme**: CCG supertags, random tags, same tags.

Hence, we explore 14 combinations (2 architecture baselines + 12 multi-task combinations) for DE-EN translation.

3.4. Implementation Details

With regards to token representation, BPE encoding (Sennrich et al., 2016) was learned from a shared vocabulary with a final subword vocabulary size of 32k and an embedding size of 512 in all architectures.

For the Seq2Seq architecture, LSTM cells of size 512 were used in the encoder and decoder, both single-layer, with Bahdanau et al. (2014) attention. For *multi-decoder* setups, we use cells of size 128 for the tag decoder. We train on batches of 32 sentences with learning rate of 1e-5 and optimize using Adam (Kingma and Ba, 2017). We ap-

[2] A closer inspection revealed that these sentences were mainly programming source code.

ply a dropout rate of 0.2 for the outputs of the embeddings, encoder, attention, and decoder.

For the Transformer architecture, we used 6 dense layers each of width 512 and multi-headed dot-product attention with 8 heads. Values for batch size, learning rate, dropout, and choice of optimizer remain the same as in Seq2Seq training.

4. Results

This section details the results of training and evaluating DE-EN translation on the 14 models explained in the previous section. In line with the recommendations by Popel and Bojar (2018), we report not only the final scores but also the full learning curves, i.e. the BLEU scores of all models on the validation set over the duration of their training. Additionally, we provide the tag accuracies as well as the final BLEU scores on the test set.

Figure 2 displays the performance evolution of all setups during training over several million steps (sentence pairs).

The baseline Seq2Seq model increases in BLEU score the most early into training. In later stages, training the model in a multi-task configuration with either CCG or random tags results in a small boost in BLEU score. Both tag schemes closely match in translation performance. However, using same tags is extremely detrimental to training. The same tag scheme underfits the training data, resulting in a reduction of BLEU score of 20 points or more.

In the Seq2Seq experiments, there is only a slight difference between CCG and random tag schemes. In the multi-decoder setup, the random tag model gains an early lead, but nearly loses it once training is complete. In interleaved, random tags keep their position. Overall, random tags perform within ~0.5 from CCG tags and beat the baseline by ~2 BLEU points.

In the Transformer experiments, the difference between the CCG and random tag schemes are a little more apparent, but opposite. As before, the random tag scheme performs slightly better throughout the training but then falls below the CCG tag model in the last epoch. This results in CCG tags having a ~1 BLEU point lead over random tags, and ~3 BLEU point lead over the baseline. It is also worth noting that, as expected, the Transformer models slightly outperform the Seq2Seq models.

Table 1 confirms the results on the test set, with CCG tags leading to the highest score except Seq2Seq Interleaved, very closely followed by random tags. The baseline falls short two or more BLEU points except Transformer multi-decoder where the difference is smaller, 1.18 BLEU from random tags.

Finally, Table 2 presents CCG tagging accuracies for all architectures and multi-task setups. The accuracies were obtained by processing the text output of multi-task systems with EasySRL. These automatic tags then served as the golden truth against which the CCG tags proposed by the multi-task model were evaluated. We thus did not face the problem of mismatching sequence length.

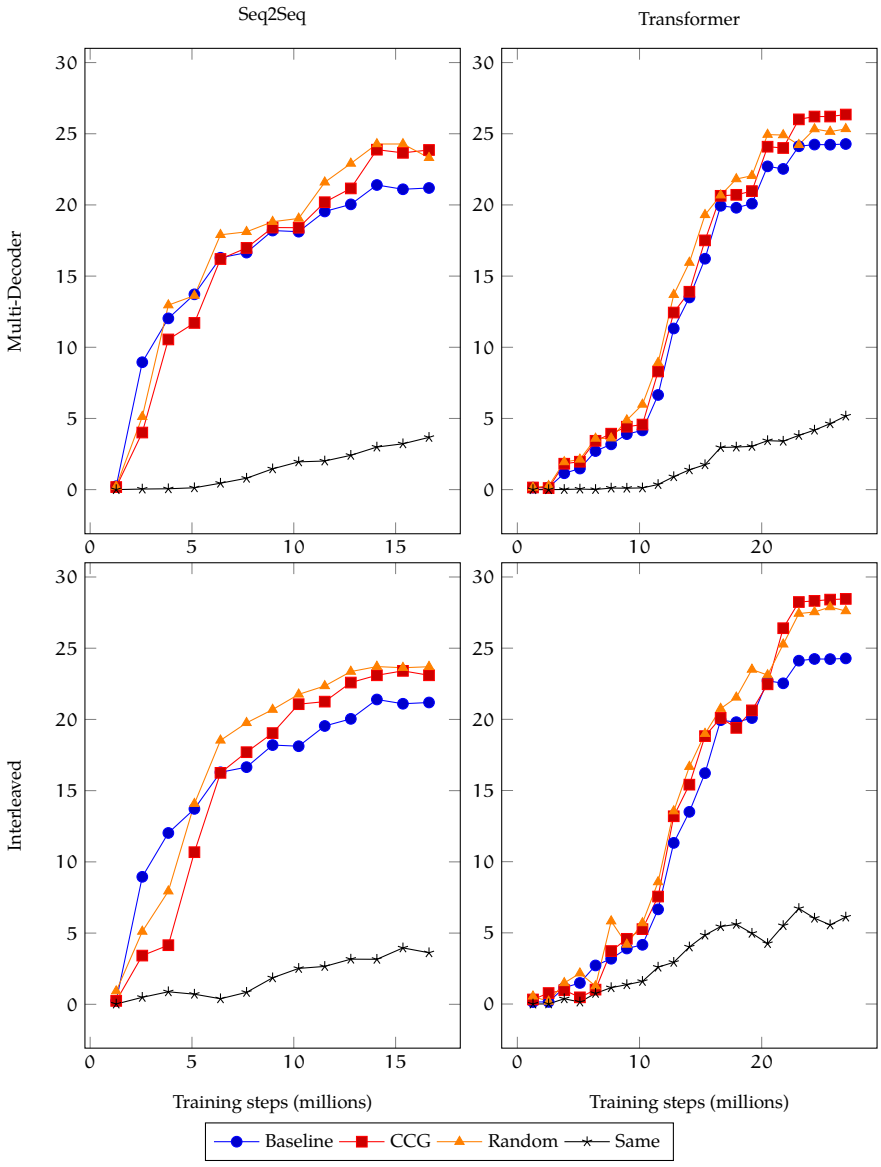

Figure 2. Performance over the DE-EN validation set according to BLEU score. Setups are organized by architecture (Seq2Seq, Transformer) and multi-task configuration (multi-decoder, interleaved), each one showing results for all tag schemes (CCG, random, and same tags). Baseline plots are repeated in both multi-task configurations for ease of comparison.

Setup	Base	CCG	Random	Same
Seq2Seq Multi-decoder	20.96	**23.66**	23.08	3.50
Transformer Multi-decoder	24.09	**26.23**	25.27	5.00
Seq2Seq Interleaved	20.96	22.96	**23.54**	3.44
Transformer Interleaved	24.09	**28.32**	27.47	5.98

Table 1. Final BLEU results on the test set. The baseline was tested for both Seq2Seq and Transformer architectures but does not include any tagging component, so it is repeated across multi-decoder and interleaved setups.

Setup	CCG Accuracy
Seq2Seq Multi-decoder	0.42
Transformer Multi-decoder	0.44
Seq2Seq Interleaved	**0.48**
Transformer Interleaved	0.39

Table 2. Final tag accuracy for all architectures and multi-task configurations, under the CCG tag scheme.

Under the random tags scheme, all setups scored 0%, while under the same tags scheme all setups got a perfect score of 100%. The reason behind this behavior can be inferred by inspecting at the accuracy over training. In all cases, it was observed that the same-tag setups quickly learn to tag all words to the same category. The random-tag models cannot learn to tag correctly, resulting in an expected accuracy of $\frac{1}{500}$ or 0.002. The CCG tag models perform better than random and learn some important relationships, but do not result in a high accuracy due to underfitting. Specifically, they reach accuracies around 40–50% when evaluated against the automatic tagging of our test set.

5. Discussion

The results indicate something surprising: predicting uncorrelated random tags in multi-task neural machine translation may perform comparably to predicting correlated, linguistically-informed, CCG tags. In other words, it is possible that the network is learning some syntactic information (as documented by reasonable performance in tagging accuracy, Table 2) but it is not utilizing it in any useful way in the main translation task. Instead, gains in translation task are obtained thanks to some changes in numerical properties of the training. This result holds even across several different neural architectures. This goes against the intuition that the network would be able to learn and benefit from a representation that generalizes over both tasks.

Maybe some joint representation is indeed learned in the multi-task setting, or maybe the two tasks live independently of each other. It is still unclear how much the CCG tags provide useful generalizations and how much they are acting as some simple regularizer. The interleaved setups probably also benefit from the increased effective depth of the decoder: while emitting the tag, the decoder can work on refining its internal state. This is particularly likely with the random tags where the network can quickly notice its zero chance of finding any pattern and reuse the additional capacity for better learning of the main task.

This primary result may also explain why multi-task neural machine translation is difficult. Other works have shown that neural networks can learn to generalize over multiple tasks. However, it is crucial that the tasks and representations of those tasks are similar enough so that the networks can infer those relationships. Otherwise, the network may not be able to reconcile the two representations, which the above experiments may also suggest.

The main message we would like to express is that multi-task experiments should always consider baseline runs with dummies, to validate that the improvements are from the secondary task and not from simple regularization or other unintended effects, not related to the added knowledge.

6. Conclusion

Our experiments have shown that a neural machine translation model in a multi-task tagging configuration is able to perform nearly as well on uncorrelated random tags as on true CCG tags. This casts doubt on the intuition that improvements observed in previous works in multi-task neural models with syntactic information are in all cases due to the model's improved generalization over syntax.

As a result, we propose future multi-task neural machine translation experiments should include trivial baseline experiments where the secondary tasks are replaced with random data to ensure that the knowledge of the secondary task is indeed crucial for the observed improvements. More experimentation is necessary to determine in what cases multi-task neural models can generalize and what cases these models interpret secondary tasks as random noise.

Acknowledgements

This study was supported in parts by the grants H2020-ICT-2018-2-825303 (Bergamot) of the European Union and 19-26934X (NEUREM3) of the Czech Science Foundation.

Bibliography

Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

Bentivogli, Luisa, Arianna Bisazza, Mauro Cettolo, and Marcello Federico. 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, 2016.

Bradbury, James and Richard Socher. Towards Neural Machine Translation with Latent Tree Attention. In *Proceedings of the 2nd Workshop on Structured Prediction for Natural Language Processing*, pages 12–16, 2017.

Dyer, Chris, Adhiguna Kuncoro, Miguel Ballesteros, and Noah A Smith. Recurrent Neural Network Grammars. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 199–209, 2016.

Eriguchi, Akiko, Kazuma Hashimoto, and Yoshimasa Tsuruoka. Tree-to-Sequence Attentional Neural Machine Translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 823–833, 2016.

Eriguchi, Akiko, Yoshimasa Tsuruoka, and Kyunghyun Cho. 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)*, volume 2, pages 72–78, 2017.

Helcl, Jindřich and Jindřich Libovický. Neural Monkey: An Open-source Tool for Sequence Learning. *The Prague Bulletin of Mathematical Linguistics*, (107):5–17, 2017. ISSN 0032-6585. doi: 10.1515/pralin-2017-0001. URL `http://ufal.mff.cuni.cz/pbml/107/art-helcl-libovicky.pdf`.

Hockenmaier, Julia and Mark Steedman. CCGbank: a corpus of CCG derivations and dependency structures extracted from the Penn Treebank. *Computational Linguistics*, 33(3):355–396, 2007.

Kingma, Diederik P and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2017.

Koehn, Philipp, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. Moses: Open Source Toolkit for Statistical Machine Translation. In *ACL 2007, Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic, June 2007. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/P/P07/P07-2045`.

Lewis, Mike, Luheng He, and Luke Zettlemoyer. Joint A* CCG parsing and semantic role labelling. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1444–1454, 2015.

Nadejde, Maria, Siva Reddy, Rico Sennrich, Tomasz Dwojak, Marcin Junczys-Dowmunt, Philipp Koehn, and Alexandra Birch. Predicting Target Language CCG Supertags Improves Neural Machine Translation. In *Proceedings of the Second Conference on Machine Translation*, pages 68–79, 2017.

Niehues, Jan and Eunah Cho. Exploiting Linguistic Resources for Neural Machine Translation Using Multi-task Learning. In *Proceedings of the Second Conference on Machine Translation*, pages 80–89, 2017.

Papineni, Kishore, Salim Roukos, Todd Ward, and Wei-Jing Zhu. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics, 2002.

Popel, Martin and Ondřej Bojar. Training Tips for the Transformer Model. *The Prague Bulletin of Mathematical Linguistics*, 110(1):43–70, 2018. URL `https://content.sciendo.com/view/journals/pralin/110/1/article-p43.xml`.

Sennrich, Rico, Barry Haddow, and Alexandra Birch. 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)*, volume 1, pages 1715–1725, 2016.

Shi, Xing, Inkit Padhi, and Kevin Knight. Does string-based neural MT learn source syntax? In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1526–1534, 2016.

Steedman, Mark. The syntactic process. 2000.

Tai, Kai Sheng, Richard Socher, and Christopher D Manning. Improved semantic representations from tree-structured long short-term memory networks. *arXiv preprint arXiv:1503.00075*, 2015.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 6000–6010, 2017.

Address for correspondence:
Ondřej Bojar
`bojar@ufal.mff.cuni.cz`
Institute of Formal and Applied Linguistics
Malostranské náměstí 25 , Prague, 180 00, Czech Republic