Enriching the Transformer with Linguistic Factors for Low-Resource Machine Translation

Jordi Armengol-Estapé, Marta R. Costa-jussà, Carlos Escolano
TALP Research Center, Universitat Politècnica de Catalunya, Barcelona
jordi.armengol.estape@gmail.com,{marta.ruiz,carlos.escolano}@upc.edu

Abstract

Introducing factors, that is to say, word features such as linguistic information referring to the source tokens, is known to improve the results of neural machine translation systems in certain settings, typically in recurrent architectures. This study proposes enhancing the current state-of-the-art neural machine translation architecture, the Transformer, so that it allows to introduce external knowledge. In particular, our proposed modification, the Factored Transformer, uses linguistic factors that insert additional knowledge into the machine translation system. Apart from using different kinds of features, we study the effect of different architectural configurations. Specifically, we analyze the performance of combining words and features at the embedding level or at the encoder level, and we experiment with two different combination strategies. With the best-found configuration, we show improvements of 0.8 BLEU over the baseline Transformer in the IWSLT German-to-English task. Moreover, we experiment with the more challenging FLoRes English-to-Nepali benchmark, which includes both extremely low-resourced and very distant languages, and obtain an improvement of 1.2 BLEU.

1 Introduction

Many classical Natural Language Processing (NLP) pipelines used linguistic features (Koehn and Hoang, 2007; Du et al., 2016). In recent years, the rise of neural architectures has diminished the importance of the aforementioned features. Nevertheless, some works have still shown the effectiveness of introducing linguistic information into neural machine translation systems, typically in recurrent sequence-to-sequence (Seq2seq) architectures (Sennrich and Haddow, 2016; García-Martínez et al., 2016; España-Bonet and van Genabith, 2018). By factored Neural Machine Translation (NMT), we refer to the use of word features alongside the words themselves to improve translation quality. Both the encoder and the decoder of a Seq2seq architecture can be modified to obtain better translations (García-Martínez et al., 2016). The most prominent approach consists of modifying the encoder such that instead of only one embedding layer, the encoder has as many embedding layers as factors, one for words themselves and one for each feature, and then the embedding vectors are concatenated and input to the rest of the model, which remains unchanged (Sennrich and Haddow, 2016). The embedding sizes are set according to the respective vocabularies of the features. Note that they used Byte Pair Encoding (BPE) (Sennrich et al., 2016), an unsupervised preprocessing step for automatically splitting words into subwords with the goal of improving the translation of rare or unseen words. Thus, the features had to be repeated for each subword.

In España-Bonet and van Genabith (2018), the exact same architecture was used, except that this new proposal used concepts extracted from linked data database, BabelNet (Navigli and Ponzetto, 2012). These semantic features, synsets, were shown to improve zero-shot translations. All the cited works obtained moderate improvements with respect to the BLEU scores of the corresponding baselines.

Some works have previously proposed additional ways to combine sources and introduce hierarchical linguistic information (Currey and Heafield, 2019, 2018; Libovický et al., 2018; Tebbifakhr et al., 2018).

The main goal of this work, and differently from previous works using NMT architectures based on recurrent neural networks, is to modify the Transformer to make it compatible with factored NMT with an architecture that we call Factored Transformer and inject linguistic knowledge and concepts extracted from linked data, BabelNet (Nav-

Proceedings of Recent Advances in Natural Language Processing, pages 73–78
Sep 1–3, 2021.
https://doi.org/10.26615/978-954-452-072-4_009

igli and Ponzetto, 2012). We focus on low-resource datasets.

2 Factored Transformer

Unlike the vanilla Transformer (Vaswani et al., 2017), the Factored Transformer can work with factors; that is, instead of just being input the original source sequence, it can work with an arbitrary number of feature sequences. Those features can be injected at embedding-level, as in the previous works we described above (but in a Transformer instead of a recurrent-based seq2seq architecture), or at the encoder level.

1-encoder model (depicted in Figure 1, top): Each factor, including the words themselves, has its own embedding layer. The embedding vectors of the different factors are combined, positional encoding is summed and input to the following layer. The rest of the model remains unchanged. The positional encoding is summed to the combined vector and not to each individual embedding because we are not modifying the length of the sequence; therefore, the relative positions remain unchanged.

N-encoders model (depicted in Figure 1, bottom): We intuited that features with large vocabulary sizes could benefit from having a specific encoder. In this variant, each factor has its own full encoder (instead of just its own embedding layer). The outputs from the encoder are combined and input to the following layer. The rest of the model remains unchanged.

Once we have the outputs of the multiple embedding layers (the 1-encoder) or the N-encoders, they must be aggregated before being input to the next layer. We have considered two combination strategies:

Concatenation: The outputs of the different embedding layers or encoders are concatenated.

Summation: The outputs of the different embedding layers or encoders are summed.

In both cases, the dimensions must agree. The decoder embedding size must be equal to the encoder embedding size. If the outputs from the different encoders or embedding layers are concatenated, they do not need to have the same embedding size, but the resulting embedding size is increased. Instead, if they are summed, they must share the same dimensionality, but the resulting vector size is not increased.

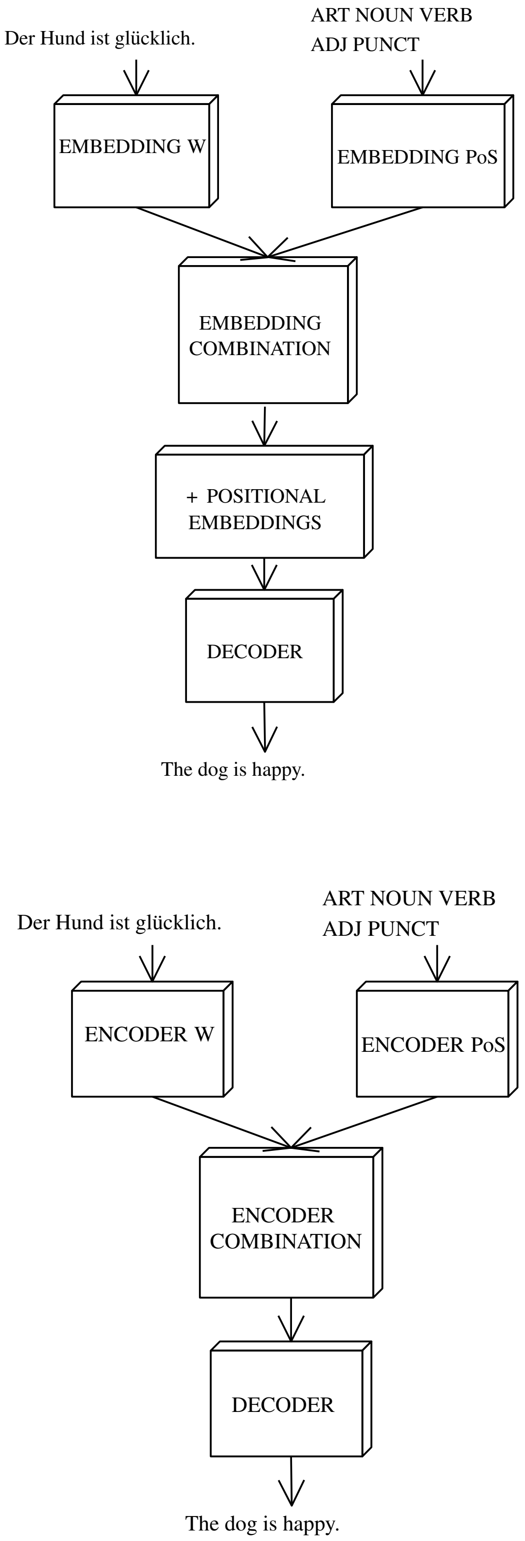

Figure 1: 1-encoder and N-encoders models.

IWSLT14			
MODEL	COMB.⋆	FEATURE	BLEU
Baseline	-	-	34.08
Lemmas	-	-	29.83
1-encoder	Sum	Lemmas	**34.35**
1-encoder	Sum	Babelnet	33.66
1-encoder	Concat	Lemmas	27.10
N-encoders	Concat	Lemmas	33.58
N-encoders	Sum	Lemmas	9.71
IWSLT16			
Baseline	-	-	36.67
1-encoder	Sum	Lemmas	**37.46**
FLoRES			
Baseline	-	-	3.06
1-encoder	Sum	Lemmas	**4.27**

Table 1: BLEU results. In bold, best results.

3 Linguistic Features

An arbitrary number of features can be injected into the Factored Transformer, provided they are aligned with words. As follows we describe how linguistic features were extracted and how they were aligned at the subword level.

Linguistic tagging with StanfordNLP: The corpus was tagged with linguistic information, namely lemmas, part-of-speech (PoS), word dependencies and morphological features, using StanfordNLP (Qi et al., 2018), and aligned with respect to the original tokenization.

Synsets extraction: BabelNet's API retrieves all possible *synsets* (semantic identifiers) that a given token may have. Babelfy (Moro et al., 2014) is a word sense disambiguation service based on BabelNet that retrieves the disambiguated synset for each token depending on the sentence-level context. We split the corpus into chunks such that the daily usage limits of the API were not exceeded and no sentence was split in half (because otherwise Babelfy would have missed the context).

Babelfy returns a list of all the detected synsets with their character offsets, and they must be assigned and aligned to the original tokenization of the corpus. The following step was performed to resolve multiword synset conflicts since in the case of synsets composed of more than one token, Babelfy may retrieve one individual synset for each token and a collective one. We decided to prioritize the synset with the largest number of tokens since it seemed to give the most disambiguated information (e.g. the synset *semantic network* gives more specific information than the individual synsets *semantic* and *network*). For the tokens in the corpus that do not have an assigned synset (e.g. articles or punctuation marks), we assign a backup syntactic feature, namely, part-of-speech.

Feature Alignment at the Subword-level: To obtain state-of-the-art results in NMT, subwords (typically, BPE) is usually required. This presents a challenge with regard to word features since they must be aligned with the words themselves. The following alternatives were implemented and experimented with: just repeating the word features for each subword; using the BPE symbol in word features, in the same manner this tag is used in BPE for splitting subwords; and subword tags. This last approach was used in (Sennrich and Haddow, 2016) and it consisted of repeating the word features for each subword and introducing a new factor, subword tags, to encode the position of the subword in the original word. The 4 possible tags are: B (beginning of subword), I (intermediate subword), E (end of the subword) and O (the word was not split). This approach is not compatible with the multiencoder architecture.

4 Experimental Framework and Results

Data: Experiments were conducted with a pair composed of similar languages, the German-to-English translation direction of the IWSLT14 (Cettolo et al., 2014), which is a low-resource dataset (the training set contains about 160,000 sentences). For cleaning and tokenizing, we use the data preparation script proposed by the authors of Fairseq (Ott et al., 2019). We took the test sets from the corpus released for IWSLT14 and IWSLT16. The former was used to test the best configuration, and the latter was used to see the improvement of this configuration in another set. A joint BPE (ie. German and English share subwords) of 32,000 operations is learned from the training data, with a threshold of 50 occurrences for the vocabulary.

Other experiments were conducted with the English-to-Nepali translation direction of the FLoRes Low Resource MT Benchmark (Guzmán et al., 2019). Although this pair has more sentences than the previous one (564,000 parallel sentences), it is considered to be extremely low-resource and far more challenging because of the lack of similarity between the involved languages. In this case, we learn a joint BPE of 5000 operations (both with an algorithm based on BPE, sentencepiece (Kudo and Richardson, 2018), as proposed by the FLoRes authors, and with the original BPE algorithm).

Parameters and Configurations: In the case of German-to-English, we used the Transformer architecture with the hyperparameters proposed by the Fairseq authors: specifically, 6 layers in the encoder and the decoder, 4 attention heads, embedding sizes of 512 and 1024 for the feedforward expansion size, a dropout of 0.3 and a total batch size of 4000 tokens, with a label smoothing of 0.1. For English-to-Nepali, we used the baseline proposed by the FLoRes authors: specifically, 5 layers in the encoder and the decoder, 2 attention heads, embedding sizes of 512 and 2048 for the feedforward expansion size and a total batch size of 4000 tokens, with a label smoothing of 0.2. In both cases, we used the Transformer architecture with the corresponding parameters we described above as the respective baseline systems, and we introduced the modifications of the Factored Transformer without modifying the rest of the architecture and parameters. As mentioned previously, linguistic features were obtained through StanfordNLP (Qi et al., 2018), except the Babelnet synsets. In the case of the latter, we found that approximately 70% of the tokens in the corpus we used did not have an assigned synset and were therefore assigned PoS.

Preliminary experiments: We experimented with BPE alignment strategies (including the approaches from section 4.2), and linguistic features extracted from Stanford tagger (lemmas, part-of-speech, word dependencies, morphological features). The preliminary experiments showed that BPE alignment strategies were not very relevant, so we adopted the alignment with BPE by repeating the word feature. In addition, we found that the most promising linguistic feature was lemmas (Sennrich and Haddow, 2016).

Reported results: We report experiments with features (lemmas and synsets), architectures (1-encoder and N-encoders systems), and combination strategies (concatenation and summation). Table 1 shows the performance of the baseline and the baseline architecture but with lemmas instead of the original words. We report how different features (lemmas or BabelNet) compare for a given architecture. Then, for the best feature, lemmas, Table 1 compares different architectures, and it is shown that the best architecture is the 1-encoder with summation. Finally, the best performing system (lemmas with a 1-encoder and summation) is evaluated in another test set, IWSLT16. The selected model is relatively efficient, because it only needs an additional embedding layer.

Once we had found that the 1-encoder Factored Transformer with summation and lemmas was a solid configuration for low-resource settings, we applied this combination the more challenging Facebook Low Resource (FLoRes) MT Benchmark. Specifically, we wanted to compare how this architecture performs against the baseline reported in the original work of this benchmark. The authors report the results before applying backtranslation and with sentencepiece, which is 4.30 BLEU. We reproduced that baseline and we got slightly better results (up to 4.38 BLEU). However, our system is designed to work with BPE, not sentencepiece, which is more challenging to align to features (since subwords coming from different words can be combined into a single token). Table 1 shows that our configuration clearly outperformed the baseline with BPE (almost 40% up), and was very close to the results with sentencepiece.

Discussion: The 1-encoder system outperforms the N-encoder one. We hypothesize that the N-encoder architecture does not give good results because a completely disentangled representation for each feature is being learned, and this is not an effective strategy for factored NMT. Therefore, it is better to combine features and words at the embedding level, not at the hidden-state level.

In the case of N-encoder with concatenation, if the linguistic features are not useful if they come from a different encoder, the decoder at least can learn to ignore them. In the case of the N-encoder architecture with sum, since the outputs from different encoders, which are potentially in very different spaces, are summed, it is tough for the decoder to interpret the vectors. In this case the decoder should learn to undo a sum, which is more difficult than just learning to ignore half of the vector (i.e., assigning low weights). In the case of the 1-encoder architecture, summation gives a much more compact representation. Summing lemmas allows the decoder layers to have a dimension of 512 (instead of doubling that, which may overfit).

Regarding the reasons why lemmas outperform synsets, we believe that the problem comes from the fact that a significant proportion of the tokens do not get a synset. Instead, we can tag all words with lemmas. Besides, the use of synsets (BabelNet) intends to help at disambiguating, but the Transformer is already good at this task (Tang et al., 2018).

5 Conclusions

We have shown that the Transformer can take advantage of linguistic features but not synsets. We conclude the best configuration for the Factored Transformer is the 1-encoder model (with multiple embedding layers) with summation instead of concatenation. For the German-to-English IWSLT task, the best configuration for the Factored Transformer shows an improvement of 0.8 BLEU, and for the extremely low-resourced English-to-Nepali task, the improvement is 1.2 BLEU.

In future work, we suggest adapting the alignment algorithm to sentencepiece by combining features coming from different words into a single feature, provided their respective subwords have been merged into a single token. We suggest investigating whether linguistic features are still useful with backtranslation too.

Acknowledgements

The authors want to specially thank Mercedes García and Cristina España for the insightful discussions. This work is supported by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No. 947657).

References

Mauro Cettolo, Jan Niehues, Sebastian Stüker, Luisa Bentivogli, and Marcello Federico. 2014. Report on the 11th iwslt evaluation campaign, iwslt 2014. In *Proceedings of the International Workshop on Spoken Language Translation, Hanoi, Vietnam*, page 57.

Anna Currey and Kenneth Heafield. 2018. Multi-source syntactic neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2961–2966, Brussels, Belgium. Association for Computational Linguistics.

Anna Currey and Kenneth Heafield. 2019. Incorporating source syntax into transformer-based neural machine translation. In *Proceedings of the Fourth Conference on Machine Translation (Volume 1: Research Papers)*, pages 24–33, Florence, Italy. Association for Computational Linguistics.

Jinhua Du, Andy Way, and Andrzej Zydron. 2016. Using BabelNet to improve OOV coverage in SMT. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 9–15, Portorož, Slovenia. European Language Resources Association (ELRA).

Cristina España-Bonet and Josef van Genabith. 2018. Multilingual semantic networks for data-driven interlingua seq2seq systems. In *Proceedings of the LREC 2018 MLP-MomenT Workshop*, pages 8–13, Miyazaki, Japan.

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

Francisco Guzmán, Peng-Jen Chen, Myle Ott, Juan Pino, Guillaume Lample, Philipp Koehn, Vishrav Chaudhary, and Marc'Aurelio Ranzato. 2019. Two new evaluation datasets for low-resource machine translation: Nepali-english and sinhala-english. *CoRR*, abs/1902.01382.

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.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Jindřich Libovický, Jindřich Helcl, and David Mareček. 2018. Input combination strategies for multi-source transformer decoder. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 253–260, Brussels, Belgium. Association for Computational Linguistics.

Andrea Moro, Alessandro Raganato, and Roberto Navigli. 2014. Entity Linking meets Word Sense Disambiguation: a Unified Approach. *Transactions of the Association for Computational Linguistics (TACL)*, 2:231–244.

Roberto Navigli and Simone Paolo Ponzetto. 2012. BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence*, 193:217–250.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*.

Peng Qi, Timothy Dozat, Yuhao Zhang, and Christopher D. Manning. 2018. Universal dependency parsing from scratch. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 160–170, Brussels, Belgium. Association for Computational Linguistics.

Rico Sennrich and Barry Haddow. 2016. Linguistic input features improve neural machine translation. *CoRR*, abs/1606.02892.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. 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.

Gongbo Tang, Mathias Müller, Annette Rios, and Rico Sennrich. 2018. Why self-attention? A targeted evaluation of neural machine translation architectures. *CoRR*, abs/1808.08946.

Amirhossein Tebbifakhr, Ruchit Agrawal, Matteo Negri, and Marco Turchi. 2018. Multi-source transformer with combined losses for automatic post editing. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 846–852, Belgium, Brussels. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.