Combining Subword Representations into Word-level Representations in the Transformer Architecture

Noe Casas **Marta R. Costa-jussà** **José A. R. Fonollosa**
Universitat Politècnica de Catalunya
{noe.casas,marta.ruiz,jose.fonollosa}@upc.edu

Abstract

In Neural Machine Translation, using word-level tokens leads to degradation in translation quality. The dominant approaches use subword-level tokens, but this increases the length of the sequences and makes it difficult to profit from word-level information such as POS tags or semantic dependencies.

We propose a modification to the Transformer model to combine subword-level representations into word-level ones in the first layers of the encoder, reducing the effective length of the sequences in the following layers and providing a natural point to incorporate extra word-level information.

Our experiments show that this approach maintains the translation quality with respect to the normal Transformer model when no extra word-level information is injected and that it is superior to the currently dominant method for incorporating word-level source language information to models based on subword-level vocabularies.

1 Introduction

Currently dominant Neural Machine Translation (NMT) architectures receive as input sequences of discrete tokens taken from fixed-size source and target token vocabularies defined a priori. Before being fed to the network, the input text is tokenized and the positions of those tokens within the vocabulary table are the actual network inputs. The granularity of the tokens in those vocabularies can range from character-level, to subword-level, to word-level.

Character-level token granularity, while allowing maximum representation ability with minimal vocabulary size for alphabet-based scripts, also delegates word formation modeling to the network and makes token sequences to be much longer than with word-based tokens.

Using word-level tokens leads to very large vocabulary sizes, especially for morphologically rich languages, where the number of surface forms per lemma is high. Large token vocabularies are impractical for the current neural architectures and hardware. It is frequent to constrain the vocabulary size to a few tens of thousand tokens, which is hardly enough to fit the number of symbols in a complete word-based vocabulary. Compositional word structures like numbers pose further problems with such a granularity level, as well as proper nouns. When word-based vocabularies are used, the vocabulary is built with the most frequent surface forms in the training data, which normally leads to degradation of translation quality.

Subword-level token granularity offers a compromise between representational power and vocabulary size, especially statistically extracted subword vocabulary strategies like Byte Pair Encoding (BPE) (Sennrich et al., 2016b).

Models with word-level token vocabularies can incorporate word-level information as extra input to the model by combining it one-to-one with the token representations. Some examples of word-level information are Part of Speech (POS) tags, syntactic dependency relationships or lemmas. In order to make use of word-level information in models with subword-level token vocabularies, a usual approach is to assign the word information to all its subwords (Sennrich and Haddow, 2016). This approach, despite improving the translation quality, introduces an information assignment mismatch.

We propose to modify the Transformer architecture (Vaswani et al., 2017) to combine the learned subword representations into word representations in the encoder block. This allows to naturally incorporate any extra word-level information directly at the level of word-level representations.

This work is structured as follows: the relevant related work is described in section 2; the proposed

Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop, pages 66–71
July 5 - July 10, 2020. ©2020 Association for Computational Linguistics

approach is described in section 3, while the experimental setup is presented in section 4 and the results are described and discussed in section 5. Finally, the conclusions are drawn in section 6.

2 Related Work

The main difficulty in profiting from word-level information in subword-based NMT architectures is the word-subword token level mismatch.

Several lines of research have studied how to combine subword-level representations into word-level information in a task-agnostic way. While the approaches by Bojanowski et al. (2017), Zhao et al. (2018) and Li et al. (2018) aim at computing pre-trained word representations, other proposals integrate the computation of the word representation in the overall NMT model, either combining information from character level, like those by Luong and Manning (2016) Costa-jussà and Fonollosa (2016), from n-gram level, like the one by Ataman and Federico (2018), or from multiple granularities like the work by Chen et al. (2018). Some other approaches like those by Wang et al. (2019) and Gu et al. (2018b) try to extend this idea to obtain multilingual *conceptual* representations from character-level representations.

Nevertheless, in all those approaches, the decoder only has access to the aggregated word-level information and not to the original subword-level information. This, while mitigating the unknown word problem, cannot handle the scenario where copying from source to target is necessary, like with unseen proper names or with compositional structures like numbers. To the best of our knowledge, this type of neural architectures that condense subword/character-level information into word-level representations have not been used for integrating extra word-level information as an additional input to the model in a translation task.

On the other hand, word level information has been injected to subword-based NMT models: Sennrich and Haddow (2016) copy the word-level linguistic information (e.g. lemma, POS tag) to each of the subwords in a word. Such information is used in an embedding and is concatenated with the subword token embedding. In this method, the subwords are also injected information about whether they are the leading subword in a word or they appear in the middle of a sequence of subwords or they are the last subword.

3 Subword to Word Transformer

In the standard Transformer architecture from Vaswani et al. (2017), the encoder applies a series of self-attention layers to the input token embeddings. The output of the encoder is then used at every layer of the decoder as key and value of the multi-head attention. In these operations, the token representations in the sequences in the source batch are masked according to the original sequence lengths in tokens.

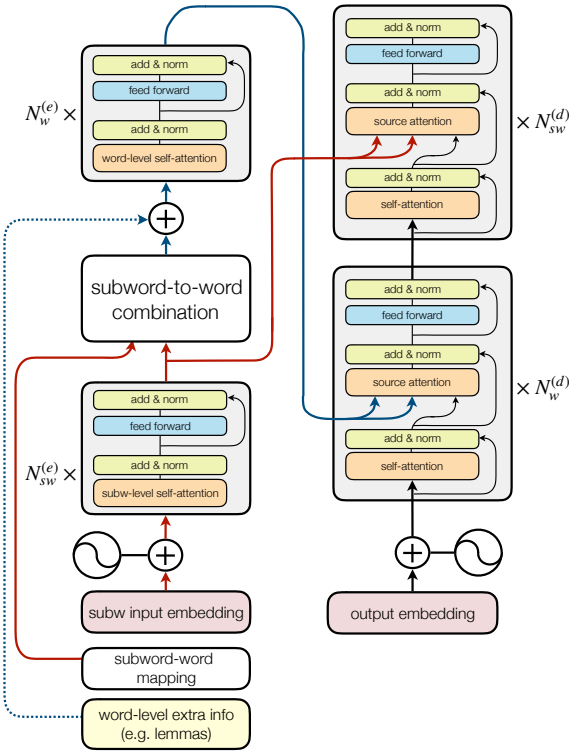

Figure 1: Subword to Word Transformer model.

We propose to divide the encoder into two blocks of self-attention layers. The first block receives the embedded subword-level token representations and processes them through $N_{sw}^{(e)}$ layers of self-attention like those from the nominal Transformer. The subword-level representations obtained as result of the first block are then combined into word level representations. A second block of $N_w^{(e)}$ self-attention layers processes these word-level representations. The output of the second encoder block is then fed to the first $N_w^{(d)}$ layers of the decoder, while the following $N_{sw}^{(d)}$ decoder layers are fed with the output of the first block of the encoder. The appropriate padding masks are used in the decoder depending on whether the encoder output used is subword or word-level. This architecture is

shown in Figure 1.

In our first tests we directly used the encoder word representations as keys and values to every decoder layer (instead of using the encoder subword representations in the last layers of the decoder). This, however, led to poor results. We understand that such a configuration made it impossible for the network to properly handle token copying from source to target, which is usually needed in cases of proper nouns or compositional structures like numbers. Other possible causes for this degradation could be some mismatch on the encoder side e.g. positional embeddings being subword-based but encoder embeddings being word-level. To test this hypothesis, we added positional encodings after the point where subword representations are combined into word-level representations. This led to no improvement, indicating that the inability to copy was certainly the cause of the degradation.

The specific approach chosen to combine subword representations into word representations is a layer of Gated Recurrent Units (GRU) (Cho et al., 2014), which receives as input the output of the first encoder block. We take the output of the GRU at the positions of the last subword tokens in each word, providing the appropriate padding positions to handle the minibatch sequences. This way, the lengths of the sequences in the batch are now the number of word tokens in each sentence.

Other subword-to-word combination approaches tested during the early stages of this work included using Long-Short Term Memories (LSTM) (Hochreiter and Schmidhuber, 1997) and simply adding all subwords within each word.

The proposed approach provides a natural point to incorporate word-level information: after the subword-level representations have been combined into word-level ones. This way, as shown in Figure 1, the extra word-level information is embedded into a vector space and added to the word-level representations of the source sentence, after the word-to-subword combination.

Note that, while applying this approach to the encoder part is straightforward, applying it to the decoder presents a key challenge: at inference time, the target side tokens are generated one by one, which implies that it is not possible to combine all of the subword tokens of a word until they have all been generated.

4 Experimental Setup

We understand that there are two desirable properties for the proposed word-subword combination model: to be able to retain the translation quality obtained with the analogous subword-based model and to be able to better profit from word-level information than other approaches.

In order to verify that the translation quality is retained, we performed experiments on the IWSLT14 English-German data, both in English→German and German→English translation directions, with a BPE shared subword vocabulary with 10K merge operations. We studied the resulting translation quality with different hyperparameter sets in order to understand their effect on the model.

In order to study the effectiveness of the proposed model with other approaches to incorporate word-level information into a subword-based model, we used the WMT16 English-Romanian data with the back-translated synthetic data from (Sennrich et al., 2016a), using a shared subword vocabulary of 40k merge operations.

We used the proposal by (Sennrich and Haddow, 2016) as baseline, and compared it to a vanilla Transformer baseline and to our proposed method.

For all experiments, we used the `fairseq` library (Ott et al., 2019), either with its built-in models for the baselines or with custom model implementations for the approach by Sennrich and Haddow (2016) and for our own proposed architecture.

For the IWSLT14 de-en and en-de baselines we used the Transformer architecture (Vaswani et al., 2017) with the hyperparameters proposed by the `fairseq` authors[1], namely 6 layers in encoder and decoder, 4 attention heads, embedding size of 512 and 1024 for the feedforward expansion size, together with dropout of 0.3 and a total batch size of 4000 tokens, using label smoothing of 0.1. For the WMT16 en-ro baseline we used the base configuration of the Transformer model offered in `fairseq`, that is, 6 layers in encoder and decoder, 8 attention heads, embedding size of 512 and 2048 for the feedforward expansion size, together with dropout of 0.1 and total batch size of 32000 tokens, without label smoothing (following the baseline used by Gu et al. (2018a)).

All reported BLEU scores are computed with the model weights averaged over the last 10 checkpoints after training until convergence.

[1] https://github.com/pytorch/fairseq/tree/master/examples/translation

5 Results

We studied the effect of different hyperparameter values over translation quality. We measured the results obtained on the IWSLT14 de-en data by using different types of subword combination strategies, as well as combining subwords at different layer levels, chosen arbitrarily. Table 1 shows how the subword combination strategy that obtains best results is to use GRU units that receive the subwords as input and return the outputs at the positions of the final subword in each word. The difference with the other alternatives is minimal, though. The rest of the hyperparameters are the same as the IWSLT14 baseline, with a total batch size of 12000 and the subword merging layers being $N_{sw}^{(e)} = 3$ and $N_{sw}^{(d)} = 3$.

Combination	BLEU
Addition	33.93
GRU	34.02
LSTM	33.92

Table 1: BLEU scores on IWSLT14 German-English for different subword combination strategies.

Regarding the influence over the translation quality of the level at which subword representations are merged, Table 2 shows that the best results are obtained when merging subwords after the fifth encoder layer, and using again the subword representations in the decoder after the third layer. The rest of hyperparameters are the same as the IWSLT14 baseline, with a total batch size of 12000 and GRU as subword combination strategy.

$N_{sw}^{(e)}$	$N_{sw}^{(d)}$	BLEU
3	5	33.53
3	3	34.02
5	3	34.46

Table 2: BLEU scores on the IWSLT14 German-English test set for different values of $N_{sw}^{(e)}$ and $N_{sw}^{(d)}$, using GRU as subword combination strategy.

Once determined that using GRU as subword combination and setting $N_{sw}^{(e)} = 5$ and $N_{sw}^{(d)} = 3$ is the hyperparameter configuration that gives the best results, we checked whether the proposed architecture maintains the translation quality with respect to a vanilla Transformer baseline. As shown in Table 3, the BLEU scores are practically the same for both architectures and both German→English while for English→German there is a small decrease. As commented in section 4, the baseline uses a batch size of 4000 while our approach uses 12000. Note that for the baseline architecture, larger batch sizes actually decrease the resulting translation quality.

	en-de	de-en
Base Transformer	28.75	34.44
Word-subword model	28.29	34.46

Table 3: BLEU scores on the IWSLT14 German-English data, using no extra word-level information.

Finally, in order to assess our proposed approach at incorporating extra word-level information, we compared it against the approach by Sennrich and Haddow (2016) (with the Transformer as base architecture), which copies the word level information to each of the subwords in the word; in our implementation, the subword embedding and the linguistic information are combined by adding them together, which is analogous to the original alternative that concatenates them. For the vanilla Transformer and the approach by Sennrich and Haddow (2016) we used a total batch size of 32000 while for the word-subword model (our proposal), we used a total batch size of 40000, GRU as subword combination strategy and $N_{sw}^{(e)} = 5$ and $N_{sw}^{(d)} = 3$.

	en-ro
Base Transformer	27.02
Word-level info copied to subwords	27.29
Word-subword model + word-level info	27.82

Table 4: BLEU scores measured on the WMT16 English-Romanian data, with lemmas as linguistic info.

The word-level linguistic information used was only the lemma (using a vocabulary of 40k lemmas), which is the feature that should provide the largest improvement according to Sennrich and Haddow (2016). We used Stanford CoreNLP (Manning et al., 2014) to annotate the corpus with the English lemmas. The obtained results are shown in Table 4, where our proposed approach obtains the best BLEU score compared to the base Transformer model (Vaswani et al., 2017) without any word-level information, and to copying the word-level info to subwords (Sennrich and Haddow, 2016).

6 Conclusion

In this work, we proposed a modification to the Transformer architecture to merge the subword representations from the first layers of the encoder into word-level representations. Merging word-level representations inside the model allows it to use the subword-level representations in the final decoder layers so that it can handle compositional structures and other situations where copying from source is needed. This approach provided an appropriate point to incorporate linguistic word-level information and it is superior at doing so compared with the reference approach by Sennrich and Haddow (2016).

Future extensions to this work may include applying it to character-level instead of subword representations, and using it for morphologically richer languages, especially low-resourced agglutinative ones, where our approach, together with the incorporation of linguistic information, may provide larger improvements in translation quality. Further extensions may include studying the behavior of more powerful subword combination strategies (e.g. convolutions, self-attention) and the application of subword merging to the target side.

Acknowledgements

This work is partially supported by Lucy Software / United Language Group (ULG) and the Catalan Agency for Management of University and Research Grants (AGAUR) through an Industrial PhD Grant. This work is also supported in part by the the Spanish Ministerio de Economía y Competitividad, the European Regional Development Fund through the postdoctoral senior grant Ramón y Cajal and by the Agencia Estatal de Investigación through the project EUR2019-103819.

References

Duygu Ataman and Marcello Federico. 2018. Compositional representation of morphologically-rich input for neural machine translation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 305–311, Melbourne, Australia. Association for Computational Linguistics.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Huadong Chen, Shujian Huang, David Chiang, Xinyu Dai, and Jiajun Chen. 2018. Combining character and word information in neural machine translation using a multi-level attention. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1284–1293, New Orleans, Louisiana. 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.

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.

Jiatao Gu, James Bradbury, Caiming Xiong, Victor O.K. Li, and Richard Socher. 2018a. Non-autoregressive neural machine translation. In *International Conference on Learning Representations*.

Jiatao Gu, Hany Hassan, Jacob Devlin, and Victor O.K. Li. 2018b. Universal neural machine translation for extremely low resource languages. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 344–354, New Orleans, Louisiana. Association for Computational Linguistics.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Bofang Li, Aleksandr Drozd, Tao Liu, and Xiaoyong Du. 2018. Subword-level composition functions for learning word embeddings. In *Proceedings of the Second Workshop on Subword/Character Level Models*, pages 38–48, New Orleans. Association for Computational Linguistics.

Minh-Thang Luong and Christopher D. Manning. 2016. Achieving open vocabulary neural machine translation with hybrid word-character models. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1054–1063, Berlin, Germany. Association for Computational Linguistics.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60.

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*.

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. Edinburgh neural machine translation systems for WMT 16. In *Proceedings of the First Conference on Machine Translation*, pages 371–376, Berlin, Germany. Association for Computational Linguistics.

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, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*.

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 I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 6000–6010. Curran Associates, Inc.

Xinyi Wang, Hieu Pham, Philip Arthur, and Graham Neubig. 2019. Multilingual neural machine translation with soft decoupled encoding. In *International Conference on Learning Representations*.

Jinman Zhao, Sidharth Mudgal, and Yingyu Liang. 2018. Generalizing word embeddings using bag of subwords. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 601–606, Brussels, Belgium. Association for Computational Linguistics.