Incorporating Hypernym Features for Improving Low-resource Neural Machine Translation

Abhisek Chakrabarty, Haiyue Song, Raj Dabre,
Hideki Tanaka, and **Masao Utiyama**
National Institute of Information and Communications Technology
3-5 Hikaridai, Seika-cho, Soraku-gun, Kyoto, 619-0289, Japan
{abhisek.chakra,haiyue.song,raj.dabre,hideki.tanaka,
mutiyama}@nict.go.jp

Abstract

Parallel data is difficult to obtain for low-resource languages in machine translation tasks, making it crucial to leverage monolingual linguistic features as auxiliary information. This article introduces a novel integration of hypernym features into the model by combining learnable hypernym embeddings with word embeddings, providing semantic information. Experimental results based on bilingual and multilingual models showed that: (1) incorporating hypernyms improves translation quality in low-resource settings, yielding +1.7 BLEU scores for bilingual models, (2) the hypernym feature demonstrates efficacy both in isolation and in conjunction with syntactic features, and (3) the performance is influenced by the choice of feature combination operators and hypernym-path hyperparameters.

1 Introduction

Low-resource neural machine translation (NMT) is an open challenge to NLP researchers because of a number of bottlenecks, such as a lack of parallel data and efficient linguistic tools, out-of-domain data, and morphological complexity of the languages. The majority of the research in this field either exploits monolingual and multilingual data in different ways (including back-translation (Edunov et al., 2018), transfer learning (Nguyen and Chiang, 2017; Song et al., 2020), and multilingual training (Dabre et al., 2020)) or come up with model-centric techniques for better modeling, training and inference (Haddow et al., 2022). Other than these two approaches, the use of linguistic knowledge is an effective strategy to improve translation quality under resource-scarce situations, however, relatively under-explored.

Linguistic analysis can be utilized for NMT both implicitly and explicitly. Implicit integration refers to methods that, instead of directly applying morphology into the model, use it as a part of pre-processing (subword segmentation of words based on legitimate units (Sánchez-Cartagena et al., 2020) or make a better contextual representation of the source sentence with the help of its syntactic/dependency structure (Eriguchi et al., 2016; Li et al., 2018; Bugliarello and Okazaki, 2020). In case of explicit use, either morphological information is included in the data to provide richer information about the source and the target languages (Sennrich and Haddow, 2016) or the model is trained with a multi-task objective to predict words along with their linguistic properties as secondary output (Luong et al., 2015) in order to obtain better internal word-form representation.

While morphological attributes are directly used in the source side as additional input features to words, it is hard to decide which input feature(s) are optimum to feed to the model for learning source-to-target mapping. Since the features are embedded in continuous space and can be combined easily, existing studies (Sennrich and Haddow, 2016; Chakrabarty et al., 2020; Chakrabarty et al., 2022) are found to use the following attributes together as supplementary components of a word - (1) part-of-speech (POS): tells syntactic behavior of individual words, (2) lemma: denotes base form and help to disambiguate inflectional variants, and (3) dependency parsing label:

Proceedings of the First International Workshop on Knowledge-Enhanced Machine Translation, pages 1–6
June 27, 2024 ©2024 European Association for Machine Translation

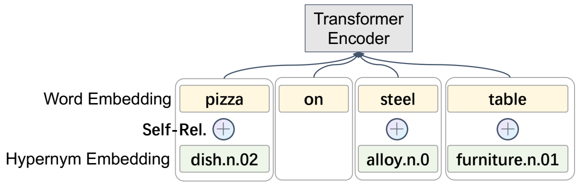

Figure 1: Incorporating hypernym features in the Encoder through Self-Relevance operation.

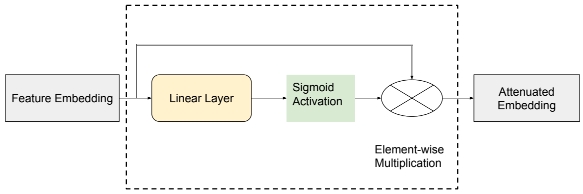

Figure 2: Self-relevance of a feature embedding.

provides the relationship with other words within a sentence. Although these three attributes are empirically established to help low-resource translation, however, they cannot impart distributional semantics which is crucial when there is not a sufficient amount of data available to learn linguistic regularities.

In this work, we try to address the above issue by incorporating hypernym information as a component of source words to meet the lack of distributional semantics in low-resource scenarios. As presented in Figure 1, hypernym provides superclass information, hence it can relate two distinct words with semantic similarity to some extent (e.g., *table* and *chair* in *furniture* sense) despite without any syntactic relation between them. One can argue that hypernym is an expensive knowledge typically obtained from WordNet (Miller, 1995) thus hardly available for low-resource languages. Nevertheless, building a primitive WordNet with hypernym relation is relatively easy and we aim to explore the potency of superclass information in NMT. In a nutshell, our contributions are as follows: (1) We incorporate hypernyms as a semantic component of word embeddings in low-resource MT, (2) Experimental results show BLEU score improvements from English to eight diverse low-resource Asian languages for both bilingual (+1.73 on avg.) and multilingual models (+0.24 on avg.), (3) We provide comparative analysis between syntactic vs. semantic feature combinations and hypernym-path hyperparameters variants.

2 Methodology

At first we provide the basics of two important concepts - how linguistic input features are used as additional components of a word in NMT models, and measuring the relevance of a feature embedding. Next, we describe the procedure of data annotation with hypernym information.

Linguistic Input Features into NMT: Sennrich and Haddow (2016) introduced a simple but effective way to incorporate linguistic input features into a word by concatenating word embedding and feature embeddings together. This approach supports an arbitrary number of features and enables the translation model to directly incorporate linguistic knowledge. In subword-based NMT, features corresponding to a word are replicated across its subwords. Given a source sentence s, if each of its token is represented with K features, then the i^{th} token s_i can be denoted as $s_i = (s_{i1}, \ldots, s_{iK})$. Here, s_{i1} is the word or sub-word embedding, while $s_{i2}, \ldots, s_{iK}$ represent various linguistic features. For any feature type indexed by $k \in \{1, ..., K\}$, let V_k, E_k, and d_k be the vocabulary, embedding matrix and dimension of the embedded vector, respectively, with $E_k \in \mathbb{R}^{d_k \times |V_k|}$. The embedding of token s_i, denoted by e_i, can be computed as $\boldsymbol{e}_{ik} = E_k s_{ik}$, where $\boldsymbol{e}_{ik}$ is the embedding of s_{ik}. The final embedding for s_i is obtained as $\boldsymbol{e}_i = \|_{k=1}^{K} \boldsymbol{e}_{ik}$, where $\|$ signifies the concatenation operation.

Attenuating Feature Embeddings by Relevance: The above method by Sennrich and Haddow (2016) combines word and feature embeddings blindly and lacks to evaluate the functionality of a feature in terms of translation goal. Chakrabarty et al., (2020) claimed that providing extra morphological information may lead to noise when a word has only one sense. Hence, they came up with two strategies to attenuate feature embedding. The first one, named as *self-relevance*, measures the importance of a feature embedding w.r.t the embedding itself, and the second one, named as *word-based relevance*, considers both word and feature embeddings together to weight the feature embedding. Out of these two relevance mechanisms, Chakrabarty et al., (2020) empirically found self-relevance to be better. Hence, we use it throughout our experimentation and detail it as follows.

Self-Relevance: For the k^{th} feature component s_{ik}, its embedded vector $\boldsymbol{e}_{ik}$ is transformed by a learnable weight matrix $W_k \in \mathbb{R}^{d_k \times d_k}$ followed by a sigmoid activation. It generates a mask vec-

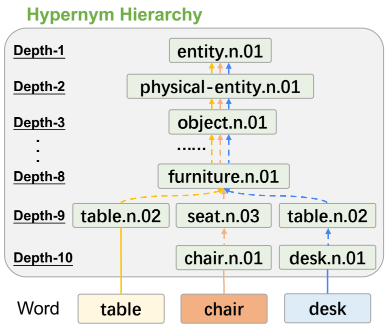

Figure 3: Depth-wise sense similarity of the synsets *table.n.02*, *chair.n.01*, and *desk.n.01*.

tor $mask_{ik}$ that contains the weight of $\boldsymbol{e}_{ik}$ as $mask_{ik} = \text{sigmoid}(W_k \boldsymbol{e}_{ik})$. Next, $\boldsymbol{e}_{ik}$ is element-wise multiplied by $mask_{ik}$ to modulate the relevance. The attenuated feature embedding is thus $\boldsymbol{e}'_{ik} = mask_{ik} \odot \boldsymbol{e}_{ik}$. Eventually all modified embeddings $\boldsymbol{e}'_{i1}, \ldots, \boldsymbol{e}'_{iK}$ are concatenated to form the source embedding $\boldsymbol{e}'_i = \|_{k=1}^{K} \boldsymbol{e}'_{ik}$ for token s_i. The process is depicted in Figure 2.

2.1 Hypernym as Additional Feature

As mentioned earlier, although there have been previous studies regarding the inclusion of several morphological attributes (POS, lemma, dependency labels, etc.) as the source word component for improving translation quality, no significant work has explored the potency of hypernym information for this purpose. Our effort is inspired by the recent work of (Bai et al., 2022) that builds a class-based language model to address context sparsity where words with common WordNet hypernyms are mapped to the same class. Inspired by this study, we hypothesize that hypernym as an input feature can alleviate the lack of distributional semantics in low-resource MT tasks.

We leverage WordNet (Miller, 1995) that defines a synset by grouping all related words together that uniquely represent one meaningful concept. It is a directed graph where nodes are synsets and the edges denote the relationships. Hypernymy conveys *[is-a]* relation between two synsets from superclass to subclass such as *furniture.n.01* $\rightarrow$ *table.n.02*. For two words, if there is a common hypernym in their respective hypernym-paths at a certain depth (from the root synset), it signifies their similarity at that depth. Figure 3 shows an example of three words 'table', 'chair', and 'desk' with their sense similarity at different depths obtained from hypernym-path information.

To annotate a word with hypernym, we follow the token-to-class mapping algorithm proposed by Bai et al., (2022) which uses the following constraints - (1) the word should have a noun synset, (2) the length of the hypernym-path should be longer than a minimum depth d, and (3) frequency of the word is less than a threshold frequency f. Bai et al., (2022) restricted to nouns because these are the most difficult class for language models to learn and hence, we also keep this constraint to annotate only those words that have at least one noun synset. A higher depth signifies deeper semantic matching and frequency filtering is applied to prevent function words. Words not satisfying the above points are tagged with *dummy* hypernym. Note that a word may present in multiple synsets corresponding to different senses and thus, it is very difficult to find the most appropriate hypernym-path for a given context. Therefore, we follow the standard strategy to iterate over the synsets in the order of sense frequency and choose the most frequent one following the depth constraint. It is safe not to set a large value of d to prevent a word annotated with inappropriate hypernym w.r.t its context.

3 Experimental Settings

Datasets: We chose Asian Language Treebank (ALT) (Riza et al., 2016) for our MT experiments, which is a multi-parallel MT dataset. The data is initially in English and translated into 12 Asian languages. Following the experimental settings of Chakrabarty et al., (2020), we fix English (en) as the source and eight Asian languages - Bengali (bg), Filipino (fi), Hindi (hi), Indonesian (id), Khmer (khm), Malay (ms), Myanmar (my) and Vietnamese (vi) as the targets. The size of the train/dev/test split for each language pair is $18,088/1,000/1,018$. In the bilingual setup, we trained eight separate NMT models for each direction, whereas in multilingual experiments, we trained a one-to-many NMT model from English to eight Asian languages. We use English as the source due to the availability of hypernyms and other morphological attributes.

Preprocessing: We apply Byte-Pair Encoding (BPE) (Sennrich et al., 2016) with 32k merge operations for subword segmentation. Multilingual setup identifies each target language by a special token appended at the source side. For English data, Stanford CoreNLP toolkit (Manning et al., 2014) is used to get POS, lemma, and dependency

ID	Features	d/f	Combination	en→bg	en→fi	en→hi	en→id	en→khm	en→ms	en→my	en→vi	Avg.
Results of Bilingual Models												
1†	-	-	-	7.50	26.98	23.62	30.88	26.24	35.78	16.48	29.05	24.57
2	H	6/6k	Self-Rel	7.51	26.63	24.09	31.23	26.54	35.49	16.91	29.76	24.77
3	H	6/50k	Self-Rel	7.45	26.85	23.56	31.09	26.36	35.62	16.93	30.02	24.74
4	H	3/50k	Self-Rel	7.39	27.26	24.35	31.52	26.63	36.34	17.70	29.38	25.07
5†	PLD	-	Self-Rel	8.40	28.22	26.13	32.65	27.33	**37.22**	18.13	29.91	26.00
6	H+PLD	6/6k	Self-Rel	8.37	28.08	25.72	**32.70**	27.90	37.19	**18.64**	31.30	26.24
7	H+PLD	6/50k	Self-Rel	**8.44**	**28.64**	**26.24**	32.68	27.83	36.80	18.46	31.29	**26.30**
8	H+PLD	3/50k	Self-Rel	8.35	28.17	26.05	32.44	**28.17**	36.71	18.52	**31.63**	26.25
9	H+PLD	6/50k	Concat	8.22	27.42	24.88	31.48	27.18	36.41	17.69	30.52	25.48
Results of Multilingual Models												
	Features	d/f	Combination	en→bg	en→fi	en→hi	en→id	en→khm	en→ms	en→my	en→vi	Avg.
10†	-	-	-	11.55	31.04	27.29	34.78	30.27	39.37	20.93	34.58	28.73
11	H	6/6k	Self-Rel	11.44	31.70	27.82	35.12	30.45	**39.95**	20.88	34.34	28.96
12	H	6/50k	Self-Rel	11.56	31.63	27.24	35.18	30.51	39.55	21.01	34.48	28.90
13	H	3/50k	Self-Rel	**11.67**	**31.77**	26.95	**35.50**	30.20	39.64	21.06	34.50	28.91
14†	PLD	-	Self-Rel	11.40	31.14	27.94	34.42	30.09	39.84	20.99	33.85	28.71
15	H+PLD	6/6k	Self-Rel	11.36	31.08	27.91	34.76	30.78	38.79	21.10	34.13	28.74
16	H+PLD	6/50k	Self-Rel	11.52	30.96	**28.52**	34.54	30.84	38.83	21.17	34.60	28.87
17	H+PLD	3/50k	Self-Rel	11.50	30.92	28.15	35.14	**31.08**	39.13	**21.30**	34.57	**28.97**
18	H+PLD	3/50k	Concat	11.22	30.60	27.47	34.50	30.68	38.86	21.13	**34.62**	28.66

Table 1: BLEU scores of bilingual and multilingual models. 'H', 'P', 'L', 'D' refer to hypernym, POS, lemma, and dependency tag, respectively. d and f refer to the minimum depth of hypernym-path and maximum word frequency, respectively. Line with † stands for results reported in (Chakrabarty et al., 2022).

tags as lexical and syntactic features,. Additionally, subword tag (Sennrich et al., 2016) is used as a positional feature for each subword.

Hypernym Annotation: We use WordNet (Miller, 1995) to annotate the data with hypernyms. As there is no straightforward way to find the optimum values of a minimum depth of the hypernym-path and threshold to maximum frequency of a word, we start with the standard combination of $d/f = 6/6k$ used by Bai et al., (2022). Next, we try with two other combinations: (1) $6/50k$: which does not restrict words based on frequency but prioritizes content words, and (2) $3/50k$: mapping distant words together, permitting shallower semantic matching. By setting $d = 6$, we get $1,502$ distinct synsets in the annotated data.

Hyperparameters and Training Details: We use Transformer-base model (Vaswani et al., 2017) with the standard set of hyperparameters of 6 layers, 8 attention-heads, $2,048$ as fully-connected-feed-forward dimension, $8,000$ warmup steps, Adam optimizer (Kingma and Ba, 2015), $4,096$ tokens as batch size. Dropout tuning is found to be sensitive and hence, varied from 0.1-0.4. The final token embedding dimension is set to 512 across all models to make the parameters comparable. Inference is done using beam size 5. BLEU score (Papineni et al., 2002) is used for evaluation.

4 Results

Table 1 presents the bilingual and multilingual translation results in the order of - (1) without any feature, (2) with hypernym as semantic feature (H), (3) with POS, lemma, and dependency tag (PLD) as lexical and syntactic features, and (4) all features together (H+PLD), with different hypernym-path hyperparameters and combination approaches.

Bilingual Models: Compared with the baseline model without using any feature, incorporating hypernyms showed up to 0.50 avg. BLEU improvement (ID 1 vs. ID 4). While using other linguistic knowledge also proves to be effective ($+1.43$ avg. BLEU comparing ID 1 with ID 5), combining hypernym with PLD yielded the best avg. BLEU score of 26.30 (ID 7). This proved that the hypernym feature is complementary to syntactic features in a bilingual setup. We performed statistical significance tests on individual language pairs between IDs - $(2, 6)$, $(3, 7)$ and $(4, 8)$ and found results statistically significant with $p < 0.05$ for all language pairs.

Multilingual Models: It is evident from Table 1 that multilingual training showed better translation quality than bilingual training across all language pairs because of knowledge sharing over eight language pairs. Chakrabarty et al., (2022) found that in a multilingual scenario, the inclusion of morphological attributes cannot improve over the base model (ID 14 vs. ID 10) as linguistic regularities are learned from the data itself. However, we distinctly observed the importance of adding hypernyms by comparing ID 10 vs. 11. A significant performance gain is observed for en→fi, en→hi, and en→ms directions with $p < 0.05$. Additionally, we did not obtain remarkable improvement when combining all features suggesting that in a multilingual setup, proving hypernym feature is the more helpful one.

Hypernym Hyperparameters: To determine the optimal d/f combination, we analyze the results where only hypernyms are used (IDs $2, 3, 4$ and $11, 12, 13$ for bilingual and multilingual setups, respectively). For bilingual models, $d/f = 3/50k$ (ID 4) produced the best avg. BLEU as well the best scores for en→fi, en→hi, en→id, en→khm, en→my, and en→my translation directions. For multilingual models, all three combinations (IDs $11, 12, 13$) performed equally well. Therefore, we further analyze where all features are used (IDs $15, 16, 17$) and find that $3/50k$ is the optimal combination for both settings, showing that shallow semantic annotation is better.

Feature Combinations: As throughout our bilingual and multilingual experiments from IDs $2-8, 11-17$, the self-relevance technique is selected for embedding combination, we further investigate the performance of simple concatenation of word and feature embeddings (Sennrich et al., 2016) and present the results in IDs 9 and 18, clearly showing the superiority of self-relevance.

Training Curves: Figure 4 shows the initial training plots for bilingual (en→khm) and multilingual models. Adding hypernyms slows training in every configuration but using PLD features speeds up the convergence. Validation perplexity becomes stable after around $8k$ batches but we continue training to note that longer training improves validation BLEU scores significantly. In our experiments, after $60k$ and $100k$ training steps for bilingual and multilingual models respectively, we did not observe further improvements in BLEU, proving that perplexity drop does not always correlate with BLEU gain.

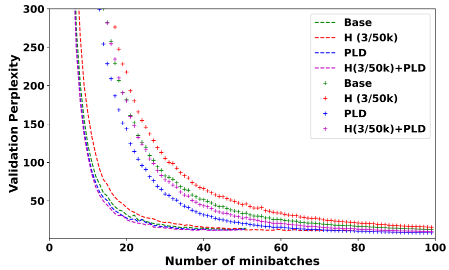

Figure 4: Bilingual (- - -) and multilingual (+) plots.

5 Conclusion

This study investigates the role of hypernyms used as a word embedding component to exploit distributional semantics in low-resource settings. Experiments over eight language pairs reveal its usefulness strongly in bilingual scenarios. We also conducted one-to-many multilingual experiments finding the superiority of semantic feature over lexical and syntactic features. We analyze training plots to show that perplexity drop is not always a good measure to evaluate model training. The future extension of this work will include - (1) finding the most appropriate hypernym-path of a contextual word, and (2) determining the optimum combination of semantic and syntactic features to leverage linguistic knowledge for low-resource translation.

References

Bai, He, Tong Wang, Alessandro Sordoni, and Peng Shi. 2022. Better language model with hypernym class prediction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1352–1362, Dublin, Ireland, May. Association for Computational Linguistics.

Bugliarello, Emanuele and Naoaki Okazaki. 2020. Enhancing machine translation with dependency-aware self-attention. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1618–1627, Online, July. Association for Computational Linguistics.

Chakrabarty, Abhisek, Raj Dabre, Chenchen Ding, Masao Utiyama, and Eiichiro Sumita. 2020. Improving low-resource NMT through relevance based linguistic features incorporation. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 4263–4274, Barcelona, Spain (Online), December. International Committee on Computational Linguistics.

Chakrabarty, Abhisek, Raj Dabre, Chenchen Ding, Hideki Tanaka, Masao Utiyama, and Eiichiro Sumita. 2022. FeatureBART: Feature based sequence-to-sequence pre-training for low-resource NMT. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 5014–5020, Gyeongju, Republic of Korea, October. International Committee on Computational Linguistics.

Dabre, Raj, Chenhui Chu, and Anoop Kunchukuttan. 2020. A survey of multilingual neural machine translation. *ACM Comput. Surv.*, 53(5), September.

Edunov, Sergey, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 489–500, Brussels, Belgium, October-November. Association for Computational Linguistics.

Eriguchi, Akiko, Kazuma Hashimoto, and Yoshimasa Tsuruoka. 2016. Tree-to-sequence attentional neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 823–833, Berlin, Germany, August. Association for Computational Linguistics.

Haddow, Barry, Rachel Bawden, Antonio Valerio Miceli Barone, Jindřich Helcl, and Alexandra Birch. 2022. Survey of low-resource machine translation. *Computational Linguistics*, 48(3):673–732, September.

Kingma, Diederik and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, San Diega, CA, USA.

Li, Qiang, Derek F. Wong, Lidia S. Chao, Muhua Zhu, Tong Xiao, Jingbo Zhu, and Min Zhang. 2018. Linguistic knowledge-aware neural machine translation. *IEEE/ACM Trans. Audio, Speech and Lang. Proc.*, 26(12):2341–2354, December.

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

Manning, Christopher D., 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.

Miller, George A. 1995. Wordnet: A lexical database for english. *Commun. ACM*, 38(11):39–41, nov.

Nguyen, Toan Q. and David Chiang. 2017. Transfer learning across low-resource, related languages for neural machine translation. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 296–301, Taipei, Taiwan, November. Asian Federation of Natural Language Processing.

Papineni, Kishore, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA, July. Association for Computational Linguistics.

Riza, Hammam, Michael Purwoadi, Gunarso, Teduh Uliniansyah, Aw Ai Ti, Sharifah Mahani Aljunied, Luong Chi Mai, Vu Tat Thang, Nguyen Phuong Thai, Rapid Sun, Vichet Chea, Khin Mar Soe, Khin Thandar Nwet, Masao Utiyama, and Chenchen Ding. 2016. https://ieeexplore.ieee.org/document/7918974/ Introduction of the Asian language treebank. In *Proc. of O-COCOSDA*, pages 1–6.

Sánchez-Cartagena, Víctor M., Juan Antonio Pérez-Ortiz, and Felipe Sánchez-Martínez. 2020. Understanding the effects of word-level linguistic annotations in under-resourced neural machine translation. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3938–3950, Barcelona, Spain (Online), December. International Committee on Computational Linguistics.

Sennrich, Rico and Barry Haddow. 2016. Linguistic input features improve neural machine translation. In *Proceedings of the First Conference on Machine Translation: Volume 1, Research Papers*, pages 83–91, Berlin, Germany, August. Association for Computational Linguistics.

Sennrich, Rico, 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, August. Association for Computational Linguistics.

Song, Haiyue, Raj Dabre, Zhuoyuan Mao, Fei Cheng, Sadao Kurohashi, and Eiichiro Sumita. 2020. Pre-training via leveraging assisting languages for neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 279–285, Online, July. Association for Computational Linguistics.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In Guyon, I., U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.