Vocabulary-informed Language Encoding

Xi Ai
College of Computer Science
University of Chonqing
barid.x.ai@gmail.com

Bin Fang
College of Computer Science
University of Chonqing
fb@cqu.edu.cn

Abstract

A Multilingual model relies on language encodings to identify input languages because it has to distinguish between the input and output languages or among all the languages for cross-lingual tasks. Furthermore, we find that language encodings potentially refine multiple morphologies of different languages to form a better isomorphic space for multilinguality. To leverage this observation, we present a method to compute a vocabulary-informed language encoding as the language representation, for a required language, considering a local vocabulary covering an acceptable amount of the most frequent word embeddings in this language. In our experiments, our method can consistently improve the performance of multilingual models on unsupervised neural machine translation and cross-lingual embedding.

1 Introduction

With tied weights across required languages, a multilingual model is trained on non-parallel or/and parallel multilingual corpora. Essentially, language encodings are required for cross-lingual tasks because the multilingual model has to distinguish between the input and output languages or among all the languages generally. We observe that, besides identifications of languages, language encodings can help the model build isomorphic space for multilinguality with the help of shared tokens. Specifically, our hypothesis derives from the decomposition of attention mechanisms (Vania and Lopez, 2017; Luong et al., 2015; Libovický and Helcl, 2017), and we observe explicit alignments and implicit alignments, where explicit alignments are key for language identifications, and implicit alignments promote multilinguality. Furthermore, the implicit alignments are a special case of unsupervised bilingual/multilingual lexical induction, helping multilingual models learn multilingual and cross-lingual knowledge. Our goal is to retain the explicit alignments for language identifications and improve the implicit alignments for multilinguality.

In this work, we render an analysis of the attention mechanism in optimization, and then we find implicit alignments among a language encoding and other language encodings. Stemming from morphology adaptation and the observation, we present a method to compute VLE (**V**ocabulary-informed **L**anguage **E**ncoding) as the required language encodings, for multilingual models. Each required language maintains a local vocabulary covering a subset of the most frequent tokens in this language from the shared vocabulary of the multilingual model. Given one language and its vocabulary, when adapting to the multilingual model, we apply transformation layers to the average of token embeddings in the vocabulary for VLE that can be used in either *padding style* or *adding style*. VLE provides language characteristics for language identification and leverages implicit alignments for multilinguility. Eventually, the multilingual model can identify languages and learn better multilinguality with the help of VLE.

2 Related Work and Background

2.1 Language Encoding

Let E_x denotes x's embedding. Given an input sentence $X = \{x_0, ..., x_n\}$ and the language encoding E_{LT_l} both in language $Lang_l$, we have: *padding style* (Johnson et al., 2017): $X_{input} = \{E_{LT_l}, E_{x_0}, ..., E_{x_n}\}$ and *adding style* (Lample and Conneau, 2019): $X_{input} = \{E_{x_0} + E_{LT_l}, ..., E_{x_n} + E_{LT_l}\}$. For notational convenience, we omit some other techniques, e.g., position encodings. Suppose we apply E_{LT_l} in *adding style* for Transform-based models[1]. For predicting x_i in the input sentence, the attention score $e_{i,j} = (E_{x_i} + E_{LT_l})^T W_q^T W_k (E_{x_j} + E_{LT_l})$ of the first self-attention layer between query vector q

[1]For the *padding style* and other models, it is similar.

Proceedings of the 29th International Conference on Computational Linguistics, pages 4883–4891
October 12–17, 2022.

and key vector k within the same sentence can be decomposed as:

$$e_{i,j} = E_{x_i}^T W_q^T W_k E_{x_j} + E_{x_i}^T W_q^T W_k E_{LT_l} + E_{LT_l}^T W_q^T W_k E_{x_j} + E_{LT_l}^T W_q^T W_k E_{LT_l}, \quad (1)$$

where W_q and W_k are transformation layers for q and k respectively.

2.1.1 Explicit and Implicit Alignment

Specifically, multilingual models (Johnson et al., 2017; Devlin et al., 2019; Lample and Conneau, 2019) usually form a vocabulary that covers shared tokens across 1+ languages. Suppose x_i is shared by $Lang_l$ and $Lang_{l'}$. In optimization, we have two backward passes from predicting x_i: *1)* $\frac{\partial \varepsilon_{x_i}}{\partial E_{LT_l}}$ and *2)* $\frac{\partial \varepsilon_{x_i}}{\partial E_{LT_{l'}}}$, that E_{LT_l} and $E_{LT_{l'}}$ are aligned to x_i explicitly. Then, we have the pivoted alignment $E_{LT_l} \leftrightarrow E_{x_i} \leftrightarrow E_{LT_{l'}}$. Since E_{x_i} is a point in the embedding space, $E_{LT_l} \leftrightarrow E_{x_i} \leftrightarrow E_{LT_{l'}}$ implies the implicit alignment $E_{LT_l} \leftrightarrow E_{LT_{l'}}$.

Meanwhile, we observe that unsupervised methods for word translation or lexical induction (Lample et al., 2018a; Artetxe et al., 2018) leverage similar implicit alignments to refine languages' morphologies. Concretely, given two subsets of N and M in $Lang_1$ and $Lang_2$ respectively from a shared vocabulary, (Lample et al., 2018a) explore an unsupervised domain-adversarial training (Ganin et al., 2016) for morphology adaptation that N and M are not parallel but cover the most frequent words in the two languages respectively. Embeddings in each sub-vocabulary are invariant to a language or a domain and serve as multiple anchors to identify the language and constrain the morphology. Then, the model considers the implicit alignment:

$$\frac{1}{N}\sum_{n\in N} E_{x_n} \leftrightarrow \frac{1}{M}\sum_{m\in M} E_{x_m}. \quad (2)$$

In multilingual models, $E_{LT_l} \leftrightarrow E_{LT_{l'}}$ could be viewed as a special case of Eq.2, where $|N| = |M| = 1$. Empirically, large $|N|$ and $|M|$ can help the model build isomorphic spaces (Lample et al., 2018a; Artetxe et al., 2017). Our method derives its motivation from this that vocabularies can be used to generate language encodings, i.e., $|N|, |M| > 1$, for improving multilinguality, as the morphology adaptation Eq.2 can be consistently improved by using large $|N|$ and $|M|$.

3 Our Approach

3.1 VLE

Following the previous idea and our observation, we present a method to generate language encodings from local vocabularies. Concretely, given a fixed size of vocabulary Voc_l formed by the monolingual tokens from the monolingual corpora in $Lang_l$, VLE (vocabulary-informed language encoding) for $Lang_l$ is defined as:

$$E_{Voc_l} = \frac{1}{|Voc_l|}\sum_{i\in Voc_l}, \quad E_{V_l} = \sigma(W^l E_{Voc_l}) \odot E_{Voc_l}, \quad (3)$$

where Voc_l is a local vocabulary for $Lang_l$ and $W_l \in \mathbb{R}^{d\times d}$. We introduce E_{V_l} to the multilingual model for the identification of $Lang_l$ in either *padding style* or *adding style*. Then, any two E_{V_l} and $E_{V_{l'}}$ can have the implicit alignment: $E_{V_l} \leftrightarrow E_{V_{l'}}$. Since both $|Voc_l| > 1$ and $|Voc_{l'}| > 1$, $E_{V_l} \leftrightarrow E_{V_{l'}}$ leverages the morphology adaptation Eq.2 ($|N|, |M| > 1$) for refining the morphologies of the languages to consistently improve isomorphic spaces. In our experiment, we justify this hypothesis on a cross-lingual embedding task and provide a t-SNE visualization (Van Der Maaten and Hinton, 2008) to show the improvement of aligning token pairs in two languages.

Meanwhile, E_{V_l} has to be able to represent the language. The backend of identification relies on the language characteristics from embeddings. Intuitively, the employment of $|Voc_l|$ provides some information for approximation, and then E_{V_l} gives the model global information (Shah and Barber, 2018; Ai and Fang, 2021a) covering language characteristics. Following this intuition, any method extracting common language characteristics from embeddings is feasible. In our preliminary experiments, we find that E_{V_l} could be obtained by applying a very shallow network to the average of the token embeddings in its vocabulary. In this work, we instantiate the model with feature contributions. Specifically, σ yields a probability of each embedding feature for language characteristics, i.e., contributions for language characteristics, similar to (Ai and Fang, 2021b) that uses σ to generate probabilities over vector elements. Statistics in §Experiment show some embedding features are significant to language characteristics with very high probabilities (≈ 1).

3.2 Formation of Voc_l

To find a local Voc_l for a required language from the shared vocabulary, we calculate the most frequent tokens in the monolingual corpora of this language and select a subset of Top-K tokens. However, some of the most frequent tokens are multilingual, which are shared by 1+ languages, i.e., numbers. Essentially, our Voc_l is expected to represent the language with less ambiguity. Inspired by (Wang et al., 2020), we score all the tokens with:

$$m(x) = \frac{C_l(x)}{C_{\neq l}(x)}, \tag{4}$$

where $C_l(x)$ and $C_{\neq l}(x)$ are the count of x in the monolingual corpora of $Lang_l$ and other languages' corpora respectively. Intuitively, $m(x)$ measures how monolingual x is, i.e., x_i with a high score is more monolingual than x_j with a low score. After scoring, we select tokens with the highest scores for Voc_l. Note that, if we consider language families, this scoring criterion is essential for our method. Specifically, some languages are closely related with lots of shared tokens in the vocabulary such as Spanish and Portuguese. The scoring method significantly mitigates the pain because Voc_l covers frequent tokens that appear most likely in $Lang_l$. On the other hand, for dissimilar languages with a minimum amount of shared tokens in the vocabulary, e.g., only sharing numbers, Voc_l is formed by frequent tokens that appear only in $Lang_l$.

3.3 Analysis and Discussion

Size of K The size of Voc_l is significant in practice. If K is too large (e.g., 10,000), it may cause memory problems on mediocre machines, then terminating training and inferring. However, too small K (e.g., 10) may not be able to approximate language information. Our empirical study shows that median K (e.g., 100) can facilitate training and substantially improve experimental results.

Impact of Tokenization Method We are interested in how the tokenization method impact the performance because it potentially affects the formation of the shared vocabulary and then Voc_l. Different tokenization methods may result in different vocabulary and Voc_l, e.g., BPE (Sennrich et al., 2016b) and word-level. However, we are aware that the impact is relevantly small given that: *1)* tokens of non-standard words could be monolingual and can be used for Voc_l; *2)* tokens in Voc_l for VLE do not necessarily have meaningful semantics because they work like anchors of languages's subspaces in the embedding space.

Efficiency Efficiency can be evaluated from two aspects: *1)* training and *2)* inferring. Since Voc_l is fixed for $Lang_l$, the only degradation of training efficiency comes from the dynamic computation of the average operation, the lookup operation, and the transformation, which are all fast. In inferring, E_{V_l} is a constant vector for a required language, which do not hurt inferring efficiency.

4 Experiment

Our code is implemented on Tensorflow 2.2 (Abadi et al., 2016) with 2 NVIDIA Titan Xp 12G GPUs. We accumulate gradients of 2 mini-batches per pre-training step. Since we have only 2 GPUs, this operation emulates 4 GPUs. All the links of datasets, libraries, scripts, and tools marked with $\diamond$ are listed in §Appendix. A preview version of the code is submitted, and we open the source code on GitHub.

4.1 Multilingual Task

See §Appendix for more details.

Unsupervised Neural Machine Translation UNMT (Lample and Conneau, 2019; Lample et al., 2018b; Song et al., 2019; Liu et al., 2020) tackles bilingual translation (Bahdanau et al., 2015; Vaswani et al., 2017) on non-parallel bilingual corpora without having access to any parallel sentence.

Cross-lingual Embedding Recall that we derive VLE from the study of domain adaptation in unsupervised word translation or lexical induction (Eq.2). To further investigate whether VLE improves the agnostic process of forming the isomorphic space, we test the MUSE$\diamond$ task (Lample et al., 2018a) with the provided test set and tools, which is used to evaluate cross-lingual embedding similarities. This test can quantitatively report how VLE refines and improves the morphologies to overlap each other for forming the isomorphic space.

4.2 Multilingual Framework

We adapt our method to XLM (Lample and Conneau, 2019) and MASS (Song et al., 2019) that can be used to pre-train a multilingual model with the objective of MLM (masked language modeling) (Devlin et al., 2019) or train the multilingual model for multilingual tasks from scratch or pre-training. All these frameworks need language encodings to recognize and flag the required languages.

4.3 Adaptation with VLE

Besides using the same frameworks, datasets and configurations, to minimize the changes for comparison, we only replace language encodings with our VLE and use the same styles (*padding* or *adding*) as the baseline models use. For all the tasks and frameworks, we apply our scoring method (Eq. 4) and select tokens in the model's vocabulary with the highest $K = 100$ scores to form Voc_l for every language, i.e., $|Voc_l| = 100$. See §Appendix for discussion about the size of $|Voc_l|$.

4.4 UNMT

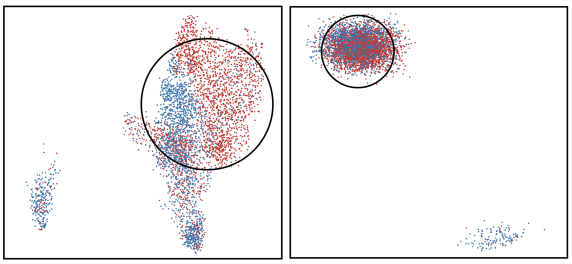
See §Appendix for more details.

Dataset For evaluation, we train UNMT on the same dataset used in previous works. We use monolingual corpora $\{Fr, De, En\}$ from WMT 2018$\diamond$ including all available $NewsCrawl$ datasets from 2007 through 2017 and monolingual corpora Ro from WMT 2016$\diamond$ including $NewsCrawl$ 2016. We test $Fr \leftrightarrow En$ on *newstest2014* and $\{De, Ro\} \leftrightarrow En$ on *newstest2016*. For any language pairs $En \leftrightarrow X$, we concatenate their monolingual corpora and then shuffle the concatenated corpus. Note that, since Ro is low-resource, we oversample Ro in pre-training and training.

Model Configuration and Preprocessing The model configuration, preprocessing, and the BLEU script are identical to previous works: XLM and MASS. Concretely, we use a 6-layer encoder and 6-layer decoder Transformer, and the dimensions of word embeddings, hidden states, and filter sizes are 1024, 1024, and 4096 respectively. All the weights and lookup tables are shared by all the required languages. We run fastBPE$\diamond$ to learn shared 60K BPE from multilingual corpora required by the multilingual model. The sampling strategy is the same as the balanced strategy presented by (Lample and Conneau, 2019). We report *case-sensitive* BLEU computed by the *multi-bleu* script $\diamond$.

4.5 Pre-training & Training

In pre-training, we use Adam optimizer (Kingma and Ba, 2015) with $\beta_1 = 0.9$,$\beta_2 = 0.999$, $\epsilon = 1e-8$, and $lr = 1e-4$. The model is pre-trained around 400K iterations. Although pre-training is important for high-performance UNMT, we also test random UNMT without pre-training to observe the lower bound and how our VLE works alone, using the MASS framework in training. In the training phase, we use Adam optimizer (Kingma and Ba, 2015) with parameters $\beta_1 = 0.9$,$\beta_2 = 0.997$ and $\epsilon = 10^{-9}$, and a dynamic learning rate with $warm\_up = 8000$ (Vaswani et al., 2017) ($learning\_rate \in (0, 7e^{-4}]$) is employed. After around 400K iterations, we report results.



(a) XLM (b) XLM+VLE

Figure 1: t-SNE visualization for MUSE pairs. Each point is a different token instance. This figure suggests that XLM aligns pairs somewhat out-of-the-box. Pairs are more aligned in-the-box when introducing VLE. Perplexity:17. Iteration:10k.

Performance Table 1 shows that our method can consistently improve the performance of baseline models on UNMT tasks, which confirms the effectiveness of our method. We observe that our method works better for low-resource Ro than for rich-resource $\{De, Fr\}$ (4% vs. 3%). As presented in *Random*, the effectiveness of our method is not from pre-training because random UNMT trained from scratch is significantly improved by using our method. Intuitively, our VLE carries multiple embeddings for morphology adaptation (recall Eq.2) that help the model understand multilinguality and cross-linguality from the isomorphic space in pre-training and training as previous works (Lample et al., 2018b; Ai and Fang, 2021b) report the effectiveness of aligning selected embeddings in UNMT. Eventually, it helps the model to learn translation knowledge. Essentially, morphology adaptation is very useful for low-resource languages.

4.6 Cross-lingual Embedding

We evaluate cross-lingual word similarities on $En \leftrightarrow De$. For our test, we use XLM and MASS that are restored by their last checkpoint of pre-training on monolingual corpora in $\{German, English\}$ from the experiments of UNMT respectively. After restoration, we extract token embeddings required by the test set via lookup tables. For words split into 2+ sub-tokens, we average all the sub-tokens. We evalu-

Model	$De \leftrightarrow En$		$Fr \leftrightarrow En$		$Ro \leftrightarrow En$	
Random *adding style*	20.99	17.12				
+ Ours	23.36	19.71				
Random *padding style*	20.86	17.08				
+ Ours	23.15	19.48				
XLM *adding style* (Lample and Conneau, 2019)	33.81	26.32	32.87	32.94	31.12	32.81
+ Ours	34.88	27.20	34.01	34.13	32.59	34.24
MASS *adding style* (Song et al., 2019)	34.91	28.03	34.42	37.02	32.75	34.82
+ Ours	35.82	28.51	35.12	37.81	34.16	36.11

Table 1: Performance of UNMT. *All the baseline models are reimplemented with our configurations. Random* denotes the model without any pre-training.

Model	MUSE (cos)
XLM(Lample and Conneau, 2019)	0.53
+VLE	0.56
MASS(Song et al., 2019)	0.55
+VLE	0.57

Table 2: Performance on MUSE task. *All the baseline models are reimplemented with our configurations.*

ate the performance by cosine similarity, reporting the result in Table 2. As expected, applying VLE can consistently improve the performance on this task, which confirms the improvements of the isomorphic space. Significantly, it confirms our hypotheses and assumptions that VLE can refine the morphologies of the languages to form a better isomorphic space. Meanwhile, we provide a t-SNE visualization (Van Der Maaten and Hinton, 2008) of the embedding space for MUSE pairs in Figure 1. This figure suggests that embedding pairs are more aligned in-the-box by using VLE.

4.7 Language characteristic

Previous works like (Ai and Fang, 2022; Conneau et al., 2020) study how different specifics of information are processed in the model, e.g., tokens and languages. As aforementioned, we expect to approximate language characteristics from $|Voc_l|$ as language encodings. Recall that, in Eq.3, we use $sigmoid$ and simply transformation layers to compute the contribution of each embedding feature for language characteristics. We present the statistics of contributions in Figure 2 from the pre-trained model on $\{En, De\}$. 10% of embedding features significantly contribute to language characteristics, obtaining over 0.8. By contrast, over 55% of embedding features are not selected for language characteristics and close to 0. These statistics can support our idea that using Voc_l and embeddings is able to provide language characteristics for language identifications. Note that, statistics may differ among different model parameters. However, the conclusions are similar in our experiments with different model parameters.

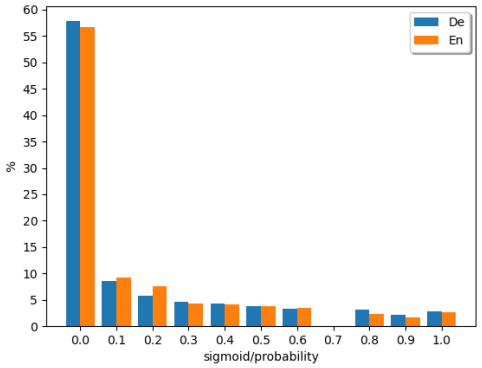


Figure 2: Contribution of each embedding feature for language characteristics (Eq.3).

4.8 Supportive Experiment

We analyze VLE on the size of $|Voc_l|$, the impact of tokenization methods, and efficiency, reporting experimental results in §Appendix.

5 Conclusion

In this work, we present a method to generate VLE, a vocabulary-informed language encoding for the identification of a required language in multilingual models. We consider a frequency-based and local vocabulary for every language. For a required language, the required language encoding is obtained by applying transformation layers to the average of the token embeddings in its vocabulary. In our experiments, VLE shows effectiveness on UNMT and cross-lingual embedding tasks and is possible to improve language adaptation and multilinguality because VLE can refine the morphologies of the languages to improve the isomorphic space. Our method is simple but effective and compatible with any other extension for multilingual models.

References

Martin Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, Manjunath Kudlur, Josh Levenberg, Rajat Monga, Sherry Moore, Derek G. Murray, Benoit Steiner, Paul Tucker, Vijay Vasudevan, Pete Warden, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2016. Tensorflow: A system for large-scale machine learning. In *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, pages 265–283.

Xi Ai and Bin Fang. 2021a. Almost free semantic draft for neural machine translation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3931–3941.

Xi Ai and Bin Fang. 2021b. Empirical regularization for synthetic sentence pairs in unsupervised neural machine translation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 12471–12479.

Xi Ai and Bin Fang. 2022. Leveraging Relaxed Equilibrium by Lazy Transition for Sequence Modeling. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, volume 1, pages 2904–2924. Long Papers.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2017. Learning bilingual word embeddings with (almost) no bilingual data. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pages 451–462, Vancouver, Canada. Association for Computational Linguistics.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2018. A robust self-learning method for fully unsupervised cross-lingual mappings of word embeddings. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pages 789–798, Melbourne, Australia. Association for Computational Linguistics.

Dzmitry Bahdanau, Kyung Hyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*.

Ond rej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Aurelie Neveol, Mariana Neves, Martin Popel, Matt Post, Raphael Rubino, Carolina Scarton, Lucia Specia, Marco Turchi, Karin Verspoor, and Marcos Zampieri. 2016. Findings of the 2016 conference on machine translation. In *Proceedings of the First Conference on Machine Translation*, pages 131–198, Berlin, Germany. Association for Computational Linguistics.

Ond rej Bojar, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, and Christof Monz. 2018. Findings of the 2018 conference on machine translation (wmt18). In *Proceedings of the Third Conference on Machine Translation*, pages 272–307, Belgium, Brussels. Association for Computational Linguistics.

Alexis Conneau, Shijie Wu, Haoran Li, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Emerging Cross-lingual Structure in Pretrained Language Models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6022–6034.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Yaroslav Ganin, Hugo Larochelle, and Mario Marchand. 2016. Domain-Adversarial Training of Neural Networks. *Journal of Machine Learning Research*, 17:1–35.

Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2017. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S Weld, Luke Zettlemoyer, and Omer Levy. 2020. Spanbert: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77.

Diederik P Kingma and Jimmy Lei Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*.

Philipp Koehn, 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. 2007. Moses: Open source toolkit for statistical machine translation. In *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. Association for Computational Linguistics.

Guillaume Lample and Alexis Conneau. 2019. Cross-lingual language model pretraining. In *Advances in neural information processing systems*.

Guillaume Lample, Alexis Conneau, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2018a. Word translation without parallel data. In *6th International Conference on Learning Representations, ICLR 2018 - Conference Track Proceedings*.

Guillaume Lample, Myle Ott, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2018b. Phrase-based & neural unsupervised machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5039–5049, Brussels, Belgium. Association for Computational Linguistics.

Jindřich Libovický and Jindřich Helcl. 2017. Attention strategies for multi-source sequence-to-sequence learning. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pages 196–202, Vancouver, Canada. Association for Computational Linguistics.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:726–742.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, Lisbon, Portugal. Association for Computational Linguistics.

Mauro Mezzini. 2018. Empirical study on label smoothing in neural networks. In *WSCG 2018 - Short papers proceedings*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 86–96, 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*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Harshil Shah and David Barber. 2018. Generative neural machine translation. In *Advances in Neural Information Processing Systems*, volume 2018-Decem, pages 1346–1355.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2019. MASS: Masked sequence to sequence pre-training for language generation. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 5926–5936. PMLR.

Laurens Van Der Maaten and Geoffrey Hinton. 2008. Visualizing Data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605.

Clara Vania and Adam Lopez. 2017. From characters to words to in between: Do we capture morphology? In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, volume 1, pages 2016–2027.

Ashish Vaswani, Google Brain, 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.

Zirui Wang, Jiateng Xie, Ruochen Xu, Yiming Yang, Graham Neubig, and Jaime Carbonell. 2020. Cross-lingual Alignment vs Joint Training: A Comparative Study and A Simple Unified Framework. In *8th International Conference on Learning Representations, ICLR 2020 - Conference Track Proceedings*.

A Experiment Setting

A.1 Multilingual Framework

We adapt our method to two MLM instances: XLM (Lample and Conneau, 2019) and MASS (Song et al., 2019), which can be used to pre-train the UNMT model. We follow the instructions of BERT (Devlin et al., 2019) and these two MLM instances to setup frameworks.

XLM XLM is similar to BERT (Devlin et al., 2019) but uses text streams of an arbitrary number of sentences. Following the instruction, we randomly select 15% of the tokens from the input sentence for replacing.

MASS MASS is different from XLM and BERT but similar to SpanBERT (Joshi et al., 2020), using spans to replace consecutive tokens. Given an input sentence with length N, we randomly select consecutive tokens with length $N/2$ for replacing.

A.2 UNMT Pipeline

We use Adam optimizer (Kingma and Ba, 2015) with $\beta_1 = 0.9$,$\beta_2 = 0.999$, $\epsilon = 1e-8$, and $lr = 1e-4$. We set the dropout regularization with a drop rate $rate = 0.1$ and label smoothing with $gamma = 0.1$ (Mezzini, 2018). For data feeding efficiency, similar-length sentences are padded to the same length, so that each mini-batch may have a different number of sentences but the same number of tokens. We pre-train the model around 400K iterations. Although pre-training is important for high-performance UNMT, we also test random UNMT without pre-training to observe

Model	$De \leftrightarrow En$	
100, default	34.88	27.20
20	34.21	26.54
50	34.72	26.96
200	35.93	27.36
500	35.08	27.51
1000	35.15	27.64

Table 3: Impact of Voc_l size on UNMT.

Model	$De \leftrightarrow En$	
baseline (BPE-based)	33.81	26.32
+ Ours	34.88	27.20
baseline (Word-level)	33.01	25.79
+ Ours	34.15	26.61

Table 4: Impact of Tokenization Method.

Model	Speed
XLM	714ms/step
+ Ours , K = 100	772ms/step
+ Ours , K = 10,000	899ms/step

Table 5: Training efficiency.

the lower bound and how our VLE works alone, using the MASS framework in training. In the training phase, we use Adam optimizer (Kingma and Ba, 2015) with parameters $\beta_1 = 0.9$,$\beta_2 = 0.997$ and $\epsilon = 10^{-9}$, and a dynamic learning rate with $warm\_up = 8000$ (Vaswani et al., 2017) ($learning\_rate \in (0, 7e^{-4}]$) is employed. We set dropout regularization with a drop rate $rate = 0.1$ and label smoothing with $gamma = 0.1$. We feed $\approx 2K$ tokens per mini-batch.After around 400K iterations, we report case-sensitive BLEU computed by *multi-BLEU.perl*$\diamond$.

We consider the same dataset used in previous works. Specifically, we first retrieve monolingual corpora $\{Fr, De, En\}$ from WMT 2018$\diamond$ (Bojar et al., 2018) including all available $NewsCrawl$ datasets from 2007 through 2017 and monolingual corpora Ro from WMT 2016$\diamond$ (Bojar et al., 2016) including $NewsCrawl$ 2016. We report the performance for $Fr \leftrightarrow En$ on *newstest2014* and $\{De, Ro\} \leftrightarrow En$ on *newstest2016*. For tokenization, we use the Moses tokenizer$\diamond$ developed by (Koehn et al., 2007). We use fastBPE$\diamond$ to learn shared 60k BPE (Sennrich et al., 2016b) with the same criteria in (Lample and Conneau, 2019).

In the pre-training phase, UNMT is trained on monolingual corpora with the objective of MLM (masked language modeling) for the two languages. Then, in the training phase, on-the-fly back-translation (Sennrich et al., 2016a) performs to generate synthetic parallel sentences that can be used for training of translation as NMT (neural machine translation) is trained on genuine parallel sentences in a supervised manner. Meanwhile, UNMT still learns the MLM objective to maintain language knowledge in the training phase.

B Supportive Result

B.1 Impact of Voc_l Size

We use $K = 100$ as the default $|Voc_l|$ for every language. In Table 3, we study the impact of $|Voc_l|$ and borrow all of the XLM configurations we use in the UNMT task. Ideally, a large $|Voc_l|$ (a significant amount of frequent tokens) can properly represent the language. However, we find a median size (< 200) is enough to achieve a decent result with minimum extra costs. Large size can achieve slightly better performance, but the computational cost is not practical, as discussed in §Size of K. In conclusion, we recommend a median size (< 200) or a finetuned size.

B.2 Impact of Tokenization Method

We are interested in how the tokenization method affects the performance because it potentially affects the formation of Voc_l. For evaluation, we use all the configurations in UNMT and additionally configure a word-level vocabulary for the model. The word-level vocabulary has the same number of tokens as the BPE vocabulary. Table 4 shows that our method can work with different tokenization methods. Our method can generally improve the performance, regardless of the difference between the two baseline models in the same configuration.

B.3 Training Efficiency

In inferring, VLE computes constant vectors for all the required languages, which do not hurt inferring efficiency. Hence, we are interested in training efficiency because we introduce some additional operations to the model. Table 5 indicates that our method does not hurt training efficiency significantly, which is crucial in applications.

C Source

We list all the links of dataset, tools, and other sources in Table 6.

Item	Links
WMT 2016	http://www.statmt.org/wmt16/translation-task.html
WMT 2018	http://www.statmt.org/wmt18/translation-task.html
XLM	https://github.com/facebookresearch/XLM
multi-BLEU.perl	https://github.com/moses-smt/mosesdecoder/blob/master/scripts/generic/multi-BLEU.perl
Moses tokenizer	https://github.com/moses-smt/mosesdecoder/blob/master/scripts/tokenizer/tokenizer.perl
fastBPE	https://github.com/glample/fastBPE
MUSE	https://github.com/facebookresearch/MUSE
Panlex	https://panlex.org/source-list/
Tensor2Tensor	https://github.com/tensorflow
HuggingFace	https://huggingface.co

Table 6: Links of source.