Coarse-To-Fine Learning for Neural Machine Translation

Zhirui Zhang[1](✉), Shujie Liu[2], Mu Li[2], Ming Zhou[2], and Enhong Chen[1]

[1] University of Science and Technology of China, Hefei, China
zrustc11@gmail.com, cheneh@ustc.edu.cn
[2] Microsoft Research Asia, Beijing, China
{shujliu,mingzhou}@microsoft.com, limugx@outlook.com

Abstract. In this paper, we address the problem of learning better word representations for neural machine translation (NMT). We propose a novel approach to NMT model training based on coarse-to-fine learning paradigm, which is able to infer better NMT model parameters for a wide range of less-frequent words in the vocabulary. To this end, our proposed method first groups source and target words into a set of hierarchical clusters, then a sequence of NMT models are learned based on it with growing cluster granularity. Each subsequent model inherits model parameters from its previous one and refines them with finer-grained word-cluster mapping. Experimental results on public data sets demonstrate that our proposed method significantly outperforms baseline attention-based NMT model on Chinese-English and English-French translation tasks.

Keywords: Neural machine translation · Coarse-to-fine learning
Hierarchical cluster

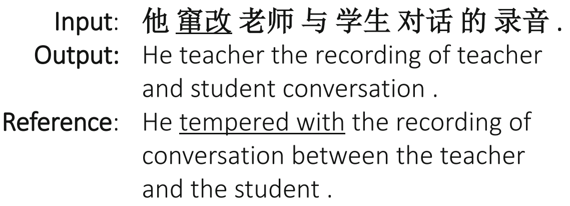
1 Introduction

As a recently proposed novel approach to machine translation, and despite its short history [2,7,14,29], neural machine translation (NMT) has been making rapid progress from catching up with statistical machine translation (SMT) [3,6,15] to outperforming it by significant margins on many language pairs [10,18,30,31,34]. Aside from better translation performance, NMT also demonstrates other appealing properties such as little requirements for human feature engineering or prior domain knowledge, so it is also drawing attention from researchers working on other NLP tasks [24,27,32].

Much recent work in the literature focuses on addressing the issue of restricted vocabulary size in NMT systems. Popular NMT system implementations employ moderate-sized vocabularies typically containing most frequent 30K–80K words, and map all the other words to a single <unk> label. Luong et al. [19] proposed a method which uses lexicon look-up to replace generated <unk> labels in target translations. This method solves part of the problem,

M. Zhang et al. (Eds.): NLPCC 2018, LNAI 11108, pp. 316–328, 2018.
https://doi.org/10.1007/978-3-319-99495-6_27

but the translation still cannot be well recovered when the unknown word rate is high, due to the fact that too many words with distinct usages sharing a single <unk> label leads to a substantial amount of ambiguities. Jean et al. [13] tackled the small vocabulary size limit with an efficient softmax approximation algorithm, which enables to use very large vocabulary in NMT systems. Although this method effectively reduces the unknown word rate and brings further improvement to translation accuracy, we note that the inclusion of more words in a larger vocabulary intensifies the challenge of learning accurate usage for the less-frequent words, even if they are not viewed as unknown words. For example, the Chinese word 窜改 (alter), which appears near the tail of a 50K-word vocabulary in terms of frequency, is such a long-tail less-frequent word. Due to its small number of occurrences in the training data, the learnt representation in a conventional NMT model is very likely to overfit to its specific usage in the training corpus, and as a result usually left ignored in unseen contexts during decoding. Figure 1 shows an incorrect translation example caused by this word.



Input: 他 窜改 老师 与 学生 对话 的 录音 .

Output: He teacher the recording of teacher and student conversation .

Reference: He tempered with the recording of conversation between the teacher and the student .

Fig. 1. Example of incorrect translation of less-frequent word.

In this paper, we present a novel NMT training method based on coarse-to-fine paradigm, which is able to learn better NMT model parameters for less-frequent words that do not have sufficient usage coverage in the training data. The presented method is inspired by a common linguistic observation that a group of words belonging to the same syntactic/semantic class, for instance, *large, enormous, gigantic, mammoth*, tend to share certain properties such as collocations and translations, and are expected to be close to each other in embedding space. This gives the opportunity that if we can assign a less-frequent word to an appropriate class whose representation can be more accurately learned, it could benefit from inheriting part of the class' representation which generalizes better to unseen contexts. Our proposed method works as follows: at first, source and target words are grouped into a set of hierarchical tree-structured clusters based on bilingual data, then a sequence of NMT models are learned based on sets of clusters at different levels of the clustering tree with finer and finer granularity. When training each model, the training data is first transformed such that all words are replaced with their corresponding clusters at the specified hierarchical level. Every cluster's representation is initialized with its parent cluster's representation learned by the previous model, then the standard NMT training process is performed to refine the model parameters.

We conduct experiments on public Chinese-English and English-French translation data sets. Experimental results demonstrate that our proposed method significantly outperforms baseline attention-based NMT model on these two translation tasks.

2 Neural Machine Translation

In this work, we concentrate on applying our coarse-to-fine learning method to sequence-to-sequence NMT models. In particular, we follow the neural machine translation architecture proposed by Bahdanau et al. [2].

Neural machine translation system is implemented as an encoder-decoder framework with recurrent neural networks (RNN), which can be Gated Recurrent Unit (GRU) [7] or Long Short-Term Memory (LSTM) [12] networks in practice. The encoder reads in the source sentence $X = (x_1, x_2, ..., x_T)$ and transforms it into a sequence of hidden states $h = (h_1, h_2, ..., h_T)$, using a bi-directional recurrent neural network. The decoder uses another recurrent neural network to generate a corresponding translation $Y = (y_1, y_2, ..., y_{T'})$ based on the encoded sequence of hidden state h. At each time i, the conditional probability of each word y_i from a target vocabulary V_y is computed by

$$p(y_i|y_{<i}, h) = g(y_{i-1}, z_i, c_i) \tag{1}$$

where z_i is the i_{th} hidden state of the decoder and is calculated conditional on the previous hidden state z_{i-1}, previous word y_{i-1} and the source context vector c_i:

$$z_i = \text{RNN}(z_{i-1}, y_{i-1}, c_i) \tag{2}$$

In attention-based NMT, the context vector c_i is a weighted sum of the hidden states $(h_1, h_2, ..., h_T)$ with the coefficients $\alpha_1, \alpha_2, ..., \alpha_T$ computed by

$$\alpha_t = \frac{\exp\left(a(h_t, z_{i-1})\right)}{\sum_k \exp\left(a(h_k, z_{i-1})\right)} \tag{3}$$

where a is a feed-forward neural network with a single hidden layer.

The whole model is jointly trained to maximize the conditional log-probability of the correct translation given a source sentence with respect to the parameters θ of the model:

$$\theta^* = \arg\max_{\theta} \sum_{n=1}^{N} \sum_{i=1}^{|y^n|} \log p(y_i^n | y_{<i}^n, x^n) \tag{4}$$

where (x^n, y^n) is the n-th training pair of sentences, and $|y^n|$ is the length of the n-th target sentence y^n.

Note that in this model, the dominant parts of the parameters θ are word embedding matrices and weight matrix for the output layer. All of them are closely related to representations of source and target words, therefore learning accurate parameters for them plays a critical role in searching for good NMT models.

3 Coarse-To-Fine Learning for NMT

Conceptually there are two major steps in our coarse-to-fine learning method: constructing a hierarchical cluster tree and learning a sequence of gradually refined NMT models. Figure 2 shows the overview framework of our approach. Based on bilingual data, a set of cluster hierarchies $\{H_0, \ldots, H_l\}$ is formed with increasing granularity and finally expands to the full vocabulary V. $M_0, \ldots, M_l$ are NMT models which use $H_0, \ldots, H_l$ as vocabularies at different level respectively and trained by bilingual data. The following of this section details how these two tasks are performed.

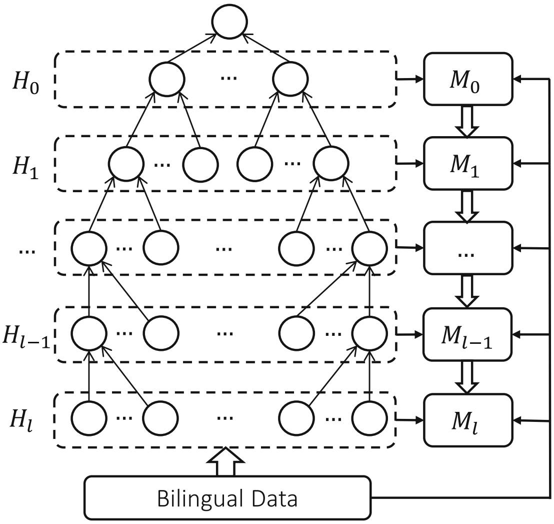


Fig. 2. The coarse-to-fine learning framework for neural machine translation.

3.1 Hierarchical Clustering

In this paper, we adopt the agglomerative hierarchical clustering algorithm to build cluster hierarchies for a given set of words.

Agglomerative hierarchical clustering algorithm works in a bottom-up manner. It starts with every word as a singleton cluster:

$$C_0 = \{a_0 = \{w_0\}, a_1 = \{w_1\}, ..., a_n = \{w_n\}\} \tag{5}$$

where C_0 is the set of initial clusters, a_i stands for cluster i, w_i denotes word in V and $n = |V|$ is the vocabulary size. Then the algorithm merges pairs of clusters step by step, until all clusters have been merged into a single cluster that contains all words. Specifically, at each step k, we have the set of clusters $C_k = \{..., a_u, ..., a_v, ...\}$. We calculate the similarity for each pair of clusters in

C_k and combine two closest clusters a_u, a_v to form a new cluster $a' = (a_u, a_v)$. The new set of clusters C_{k+1} can be represented as:

$$C_{k+1} = (C_k \setminus \{a_u, a_v\}) \cup \{a'\} \tag{6}$$

It can be easily seen that each combination reduces the number of clusters by one. So this clustering algorithm needs n steps to finish the entire procedure in total, and we have $|C_k| = n - k$.

The similarity between two clusters is measured by the cosine metric of cluster embeddings. At first, cluster embeddings in C_0 are initialized with word embeddings, which are trained from bilingual data with an improved skip-gram model proposed by Luong et al. [17]. In the following steps, the embedding of a new cluster is computed as the average of its two sub-clusters, so embedding of every cluster can be computed in a bottom-up order.

Apparently, the clustering process described above generates too many cluster sets, and it is not necessary to use all of them. Instead, before starting NMT model training, a subset of the agglomerative hierarchical clustering results needs to be selected for actual model refinement purpose.

Concretely, $H_0, \ldots, H_l$ are selected in a way that the number of clusters will grow at a geometric rate γ. Let $n_0 = |H_0|$ be the size of initial cluster H_0, H_i can be determined by the following condition

$$H_i = C_k, \quad n_0\gamma^i = |C_k| \tag{7}$$

For the last cluster set H_l, as a special case, we have $H_l = C_0 = V$ while $n_0\gamma^l \geq |C_0|$.

Note that the selected cluster sets $H_0, \ldots, H_l$ remain to be a tree structure with each cluster set H_i representing one hierarchy of the tree. For any cluster $c_p \in H_j$, there must be a parent cluster $c_q \in H_i$ satisfying $c_p \subseteq c_q$ if $j > i$.

In NMT task, the above-mentioned process is extended to support to use hierarchical clusters to refine vocabularies on both source and target side. First, we build two cluster trees, S and T for source and target words respectively, then each hierarchy of the final cluster tree is constructed by combining the corresponding hierarchy of these two cluster trees: $H_i = (S_i, T_i)$.

3.2 NMT Model Refinement

When NMT model M_{i-1} finishes training, model M_i will be learned based on the selected cluster set H_i. The learning process mostly follows the standard training procedure, but it differs from conventional NMT training in two aspects.

The first difference is the requirement for vocabulary mapping, because model M_i is expected to be trained on the vocabulary defined by H_i instead of the original vocabulary V. So a pre-processing step is needed to convert every word token in the training data into its corresponding cluster. Let (x^n, y^n) and (cx^n, cy^n) denote a word sentence pair and its cluster sentence pair respectively, and θ_i

denote the model parameters of model M_i, the objective function of NMT model training should be updated to be

$$\theta_i^* = \arg\max_{\theta_i} \sum_{n=1}^{N} \sum_{j=1}^{|y^n|} \log p(cy_j^n | cy_{<j}^n, cx^n)$$

The second difference is related to the model parameter initialization. In a conventional NMT model, all parameters are randomly initialized with some heuristics [11]. But in the coarse-to-fine learning process, only the first model M_0 is initialized in this way. All the subsequent models inherit their parameters from its previous model, that is, M_{i+1}'s parameters will be initialized with ones of M_i.

Not all parameters in model M_{i+1} can be inherited from M_i directly because their parameter structures are not fully compatible. M_{i+1} uses a larger vocabulary and thus has more parameters. Extra parameters in M_{i+1} belong to 3 categories: source word embedding, target word embedding, and weight matrix of output layer.

Our solution to this problem is to leverage the inclusion relations between clusters in H_i and H_{i+1}. The basic principle is that all sub-clusters in H_{i+1} inherit parameters of the same category from their parent cluster in H_i. Suppose $E(H_i)$ and $E(H_{i+1})$ are embedding matrices of M_i and M_{i+1}, $W_o(H_i)$ and $W_o(H_{i+1})$ denote weight matrices of output layers of M_i and M_{i+1}, and c_q is parent cluster of $\{c_{p_1}, c_{p_2}, c_{p_3}\}$. Formally, for any cluster $c_p \in H_{i+1}$, and its parent cluster $c_q \in H_i$, we have

$$E(H_{i+1})[c_p] = E(H_i)[c_q] \tag{8}$$

$$W_o^T(H_{i+1})[c_p] = W_o^T(H_i)[c_q] \tag{9}$$

Note that Eq. 8 works for both source and target clusters, while Eq. 9 is only applied to target clusters.

We notice that changing vocabulary and migrating related parameters during model transition could lead to temporary deviations in model prediction, but the deviations will be automatically fixed by later training process.

We use a validation set D to determine when to transit model learning from M_i to M_{i+1}. For each epoch during the training process, we check the perplexity change ratio ΔPPL from the last epoch: if ΔPPL is smaller than a pre-specified threshold α, the training for M_i finishes and M_{i+1} is started in the next epoch.

Algorithm 1 shows the overall training procedure. Lines 2–6 perform model initialization—except for the first model M_0, every other model is initialized with its previous model and parameter transformation function Γ defined in Eqs. 8 and 9. Word-cluster mapping is done in line 7, and lines 8–15 handle the learning of model M_i over training data T, in which α is the threshold for minimum perplexity reduction.

Algorithm 1. Coarse-To-Fine Training Algorithm for NMT

```
Input  : Bilingual data $T = \{(x^n, y^n)\}$;
         Validation set $D$;
         Cluster hierarchies $H_0, \ldots, H_l$;
Output: A sequence of NMT models $M_0, \ldots, M_l$;
for $i \leftarrow 0$ to $l$ do
    if $i == 0$ then
        Initialize $\theta_0$ in $M_0$ ;
    else
        $\theta_i = \Gamma(\theta_{i-1}, H_{i-1}, H_i)$ ;
    end
    $\{(cx^n, cy^n)\} = \mathrm{Map}(\{(x^n, y^n)\},\ H_i)$;
    for $e \leftarrow 0$ to $max\_epoch$ do
        $\theta_j^e = \arg\max_{\theta_j} \sum_T \log p(cy^n|cx^n)$ ;
        $ppl^e = \mathrm{CalcPerpelxity}(D, \theta_j^e)$ ;
        $\Delta PPL = \frac{ppl^{e-1} - ppl^e}{ppl^{e-1}}$ ;
        if $\Delta PPL < \alpha$ then
            break ;
        end
    end
end
```

4 Experiments

4.1 Setup

We evaluate our approach on two translation tasks: Chinese-English and English-French. In all experiments, we use BLEU [20] as the automatic metric for translation quality evaluation.

Dataset. For Chinese-English translation, we select our training data from LDC collection which consists of 5.2M sentence pairs with 102.1M Chinese words and 107.7M English words respectively. NIST OpenMT 2006 evaluation set is used as validation set, and NIST 2003, NIST 2005, NIST 2008 datasets as test sets.

For English-French translation, we choose a subset of the WMT 2014 training corpus used in Jean et al. [13]. This training corpus contains 12M sentence pairs with 304M English words and 348M French words. The concatenation of news-test 2012 and news-test 2013 is used as the validation set and news-test 2014 as the test set.

For each language pair, both source and target words are grouped into a cluster hierarchy respectively with agglomerative hierarchical clustering algorithm based on word embeddings. We utilize improved skip-gram model proposed by Luong et al. [17] to train word embedding on bilingual data.

Training Setting. We limit the vocabulary to contain up to 80 K most frequent words on both the source and target side, and convert remaining words into the <unk> token. In practice, we note that some of the most frequent words such as functional words, cannot gain benefit from the coarse-to-fine learning process, so we keep the 5,000 most frequent words to be singleton clusters throughout model refinement process, and all the hierarchical clustering and cluster set selection tasks are only performed on the remaining part of the vocabulary.

We adopt the RNNSearch model proposed by Bahdanau et al. [2] as our baseline, which uses a single layer GRU for encoder and decoder. The dimension of word embedding (for both source and target words) is set to 512 and the size of hidden layer is set to 1024. The matrix and vector parameters are initialized using a normal distribution with a mean of 0 and a variance of $\sqrt{6/(d_{row}+d_{col})}$, where d_{row} and d_{col} are the number of rows and columns in the structure [11]. Each NMT model is trained on a Tesla K40m GPU and optimized with the Adadelta [35] algorithm with mini-batch size set to 80. At test time, beam search is employed to find the best translation with beam size 12 and translation probabilities normalized by the length of the candidate translations. In post-processing step, we follow the work of Luong et al. [19] to handle <unk> replacement. Other hyper-parameters used in clustering and model refinement set as $\alpha = 0.05$, $n_0 = 100$ and $\gamma = 10$. In addition, we define every 1M sentences as an epoch in coarse-to-fine training process.

4.2 Results on Chinese-English Translation

Table 1 shows the evaluation results from different models on NIST datasets, in which CTF-NMT represents our coarse-to-fine methods for NMT training. In addition, we also compare our method with sub-word models - Byte Pair Encoding (BPE) [26][1]. All the results are reported based on case-insensitive BLEU.

We can observe that CTF-NMT can bring significant improvement across different test sets. These results demonstrate that coarse-to-fine training process can learn better NMT model parameters for less-frequent words so that NMT

Table 1. Case-insensitive BLEU scores (%) on Chinese-English translation. The "Average" denotes the average results of all datasets.

System	NIST2006	NIST2003	NIST2005	NIST2008	Average
RNNSearch	36.97	39.17	38.97	29.35	36.11
RNNSearch + BPE	37.58	39.73	39.87	30.48	36.92
CTF-NMT	39.14	41.69	41.02	32.66	38.63
CTF-NMT + BPE	**39.72**	**42.20**	**42.24**	**32.90**	**39.26**

[1] We learn BPE models on pre-processed source and target sentences respectively with 78K merge operations.

can yield higher quality translations. Besides, our approach achieves 1.71 points BLEU improvement than RNNSearch+BPE on average. Since BPE method splits up all words to sub-word units and expects to learn better representation for similar words that share some sub-word units, there still exist plenty syntactic or semantic similar words that do not share any sub-word units, like apple and orange. Our approach uses pre-trained word embedding to better characterize relations between these words and leverage it in NMT training, thus NMT can learn better representation for similar words. Actually, our approach also can be complementary to BPE method. We apply this method in the data preprocessed by BPE method, called CTF-NMT + BPE. In this way, another 0.63 BLEU points improvement can be achieved, which adds up to 3.15 points BLEU improvement over baseline NMT model on average. This confirms the effectiveness of combining our method with sub-word models.

4.3 Results on English-French Translation

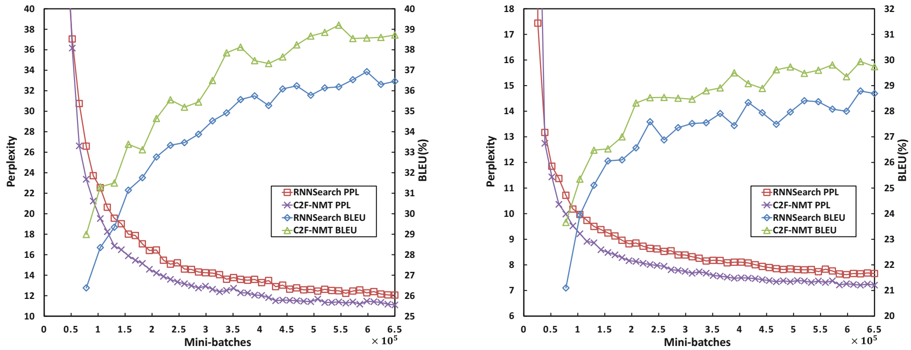
For English-French translation task, in addition to the baseline RNNSearch system, we also include results from other existing NMT systems. Experiment results are shown in Table 2. In order to be comparable with other work, all the results are reported based on case-sensitive BLEU.

First, we can see that the baseline NMT model with 80K vocabulary achieves comparable results with Jean et al. [13], which use a larger vocabulary. Also, our CTF-NMT significantly outperforms baseline NMT model with 1.34 points on test set, while achieves 0.52 points improvement than RNNSearch+BPE. When we combine our approach with BPE method, we obtain the best BLEU score 36.12 in Table 2. We believe our approach can get more improvements with deep model in future experiments.

Table 2. Case-sensitive BLEU scores (%) on English-French translation. The "PosUnk" denotes Luong et al. [19]'s technique of handling rare words. The "MRT" denotes minimum risk training proposed in Shen et al. [28]. The "LAU" represents Linear Associative Unit proposed in Wang et al. [33].

System	Architecture	Vocab Size	Test
Sutskever et al. [29]	LSTM with 4 layers	80K	30.59
Luong et al. [19]	LSTM with 6 layers + PosUnk	40K	32.70
Shen et al. [28]	Gated RNN with search + PosUnk + MRT	30K	34.23
Jean et al. [13]	Gated RNN with search + PosUnk + LV	500K	34.60
Wang et al. [33]	LAU with 4 layers	30k	35.10
Zhou et al. [37]	LSTM with 16 layers + F-F connections	30k	35.90
RNNSearch	Gated RNN with search + PosUnk	80K	34.33
RNNSearch + BPE	Gated RNN with search + BPE	80K	35.15
CTF-NMT	Gated RNN with search + PosUnk	80K	35.67
CTF-NMT + BPE	Gated RNN with search + BPE	80K	**36.12**

Figure 3 shows both perplexity and translation BLEU changes at different stages of model training for two translation tasks. To make model training with different cluster hierarchies comparable, we use word-level perplexity, which can be computed by the assumption that the probability of all words in one cluster is uniform. The BLEU is also computed at word level. We replace the generated target cluster with a word which has highest unigram probability in the cluster. From Fig. 3, it can be seen that the coarse-to-fine learning method performs consistently better (for both perplexity and BLEU) than the baseline NMT model throughout the model training process. Another observation is that, compared with the baseline system, the coarse-to-fine method needs to learn from similar amount of data (and similar training time) to achieve peak translation accuracy on validation set.



(a) Chinese-English Translation

(b) English-French Translation

Fig. 3. The perplexity (PPL) and BLEU scores on Chinese-English and English-French validation sets for RNNSearch and CTF-NMT as training progresses.

5 Related Work

This has been a long history that coarse-to-fine method is used in computer vision research, such as face detection [9] and object recognition [21]. This method has also been successfully applied to NLP tasks such as syntactic parsing [22]. Charniak et al. [4] propose a multilevel coarse-to-fine PCFG parsing algorithm, aiming at improving the efficiency of search for the best parse. Petrov et al. [23] propose a coarse-to-fine approach to statistical machine translation. They utilize an encoding-based language projection in conjunction with order-based projections to achieve speed-ups in decoding.

As a new paradigm for MT, neural machine translation has drawn more and more attention from a wide range of researchers. Resolving the OOV issue in NMT system is one of the focuses. One line of efforts [13,19] concentrated on rare words that do not exist in the system vocabulary. Jean et al. [13] explore the way based on importance sampling to directly use large vocabulary.

Luong et al. [19] propose replacement methods to handle rare words. In another direction, Costajussa et al. [8] and Sennrich et al. [26] propose character-based or subword-based neural machine translation to tackle the rare words problem. The character-based or subword-based encoding, from certain perspective, performs implicit clustering on words and affixes, and it is especially useful for morphologically rich languages such as German and Russian.

Recently, Arthur et al. [1] propose to incorporate external resources into NMT systems. Their approach employs external translation lexicons to rectify the probability distribution of rare words in the output layer. Zhang et al. [36] propose a method that leverages synthesized data to incorporate bilingual dictionaries in NMT systems, following previous work of exploiting large-scale monolingual data [5,25]. Li et al. [16] propose another method for OOV translation in NMT system: OOV words are replaced with similar in-vocabulary words during training and decoding, and the replaced words are recovered based on alignment information in decoding. Theoretically, their method can be used in in-vocabulary less-frequent words, but it is usually difficult to determine the set of words to be replaced, and requirement for accurate similar words brings more complexity to the training.

6 Conclusion

In this paper, we have presented a novel coarse-to-fine learning framework for neural machine translation. With the help of hierarchical clusters of words, our proposed method constructs a sequence of NMT models where each model refines its previous one. The key step is that each subsequent model inherits its model parameters according to cluster hierarchical relations, so that more precise representations can be learnt for less-frequent words in the vocabulary. Empirical evaluations are conducted in Chinese-English and English-French translation tasks on public available data sets. Experimental results demonstrate that our proposed method significantly outperforms baseline attention-based NMT model on these tasks.

In the future work, we plan to extend our approach to other NLP tasks and sequence-to-sequence models. Another direction we are interested in is to explore the possibility to leverage the coarse-to-fine method in incremental NMT model learning to speed-up the training process.

References

1. Arthur, P., Neubig, G., Nakamura, S.: Incorporating discrete translation lexicons into neural machine translation. In: EMNLP (2016)
2. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. CoRR abs/1409.0473 (2014)
3. Brown, P.F., Pietra, S.D., Pietra, V.J.D., Mercer, R.L.: The mathematics of statistical machine translation: Parameter estimation. Computational Linguistics (1993)
4. Charniak, E., et al.: Multilevel coarse-to-fine PCFG parsing. In: HLT-NAACL (2006)

5. Cheng, Y., et al.: Semi-supervised learning for neural machine translation. In: ACL (2016)
6. Chiang, D.: Hierarchical phrase-based translation. Computational Linguistics (2007)
7. Cho, K., et al.: Learning phrase representations using RNN encoder-decoder for statistical machine translation. In: EMNLP (2014)
8. Costa-jussà, M.R., Fonollosa, J.A.R.: Character-based neural machine translation. In: ACL (2016)
9. Fleuret, F., Geman, D.: Coarse-to-fine face detection. Int. J. Comput. Vis. **41**, 85–107 (2001)
10. Gehring, J., Auli, M., Grangier, D., Yarats, D., Dauphin, Y.: Convolutional sequence to sequence learning. In: ICML (2017)
11. Glorot, X., Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks. In: AISTATS (2010)
12. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation (1997)
13. Jean, S., Cho, K., Memisevic, R., Bengio, Y.: On using very large target vocabulary for neural machine translation. In: ACL (2015)
14. Kalchbrenner, N., Blunsom, P.: Recurrent continuous translation models. In: EMNLP (2013)
15. Koehn, P., Och, F.J., Marcu, D.: Statistical phrase-based translation. In: HLT-NAACL (2003)
16. Li, X., Zhang, J., Zong, C.: Towards zero unknown word in neural machine translation. In: IJCAI (2016)
17. Luong, T., Pham, H., Manning, C.D.: Bilingual word representations with monolingual quality in mind. In: HLT-NAACL (2015)
18. Luong, T., Pham, H., Manning, C.D.: Effective approaches to attention-based neural machine translation. In: EMNLP (2015)
19. Luong, T., Sutskever, I., Le, Q.V., Vinyals, O., Zaremba, W.: Addressing the rare word problem in neural machine translation. In: ACL (2015)
20. Papineni, K., Roucos, S.E., Ward, T., Zhu, W.J.: BLEU: a method for automatic evaluation of machine translation. In: ACL (2002)
21. Pedersoli, M., Vedaldi, A., Gonzàlez, J., Roca, F.X.: A coarse-to-fine approach for fast deformable object detection. In: CVPR (2011)
22. Petrov, S.: Coarse-to-fine natural language processing. In: Theory and Applications of Natural Language Processing (2009)
23. Petrov, S., Haghighi, A., Klein, D.: Coarse-to-fine syntactic machine translation using language projections. In: EMNLP (2008)
24. Rush, A.M., Chopra, S., Weston, J.: A neural attention model for abstractive sentence summarization. In: EMNLP (2015)
25. Sennrich, R., Haddow, B., Birch, A.: Improving neural machine translation models with monolingual data. In: ACL (2016)
26. Sennrich, R., Haddow, B., Birch, A.: Neural machine translation of rare words with subword units. In: ACL (2016)
27. Shang, L., Lu, Z., Li, H.: Neural responding machine for short-text conversation. In: ACL (2015)
28. Shen, S., et al.: Minimum risk training for neural machine translation. In: ACL (2016)
29. Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to sequence learning with neural networks. In: NIPS (2014)

30. Tu, Z., Lu, Z., Liu, Y., Liu, X., Li, H.: Modeling coverage for neural machine translation. In: ACL (2016)
31. Vaswani, A., et al.: Attention is all you need. In: NIPS (2017)
32. Vinyals, O., Kaiser, L., Koo, T., Petrov, S., Sutskever, I., Hinton, G.E.: Grammar as a foreign language. In: NIPS (2015)
33. Wang, M., Lu, Z., Zhou, J., Liu, Q.: Deep neural machine translation with linear associative unit. In: ACL (2017)
34. Wu, Y., et al.: Google's neural machine translation system: bridging the gap between human and machine translation. CoRR abs/1609.08144 (2016)
35. Zeiler, M.D.: ADADELTA: an adaptive learning rate method. CoRR abs/1212.5701 (2012)
36. Zhang, J., Zong, C.: Bridging neural machine translation and bilingual dictionaries. CoRR abs/1610.07272 (2016)
37. Zhou, J., Cao, Y., Wang, X., Li, P., Xu, W.: Deep recurrent models with fast-forward connections for neural machine translation. In: TACL (2016)