

# Multi-Reference Training with Pseudo-References for Neural Translation and Text Generation

**Renjie Zheng**<sup>1</sup>

<sup>1</sup>School of EECS, Oregon State University, Corvallis, OR

<sup>2</sup>Baidu Research, Sunnyvale, CA

zheng@renj.me {cosmmb, liang.huang.sh}@gmail.com

## Abstract

Neural text generation, including neural machine translation, image captioning, and summarization, has been quite successful recently. However, during training time, typically only one reference is considered for each example, even though there are often multiple references available, e.g., 4 references in NIST MT evaluations, and 5 references in image captioning data. We first investigate several different ways of utilizing multiple human references during training. But more importantly, we then propose an algorithm to generate exponentially many pseudo-references by first compressing existing human references into lattices and then traversing them to generate new pseudo-references. These approaches lead to substantial improvements over strong baselines in both machine translation (+1.5 BLEU) and image captioning (+3.1 BLEU / +11.7 CIDEr).

## 1 Introduction

Neural text generation has attracted much attention in recent years thanks to its impressive generation accuracy and wide applicability. In addition to demonstrating compelling results for machine translation (MT) (Sutskever et al., 2014; Bahdanau et al., 2014), by simple adaptation, practically very same or similar models have also proven to be successful for summarization (Rush et al., 2015; Nalapati et al., 2016) and image or video captioning (Venugopalan et al., 2015; Xu et al., 2015a).

The most common neural text generation model is based on the encoder-decoder framework (Sutskever et al., 2014) which generates a variable-length output sequence using an RNN-based decoder with attention mechanisms (Bahdanau et al., 2014; Xu et al., 2015b). There are many recent efforts in improving the generation accuracy, e.g., ConvS2S (Gehring et al., 2017) and

Transformer (Vaswani et al., 2017). However, all these efforts are limited to training with a single reference even when multiple references are available.

Multiple references are essential for evaluation due to the non-uniqueness of translation and generation unlike classification tasks. In MT, even though the training sets are usually with single reference (bitext), the evaluation sets often come with multiple references. For example, the NIST Chinese-to-English and Arabic-to-English MT evaluation datasets (2003–2008) have in total around 10,000 Chinese sentences and 10,000 Arabic sentences each with 4 different English translations. On the other hand, for image captioning datasets, multiple references are more common not only for evaluation, but also for training, e.g., the MSCOCO (Lin et al., 2014) dataset provides 5 references per image and PASCAL-50S and ABSTRACT-50S (Vedantam et al., 2015) even provide 50 references per image. Can we use the extra references during training? How much can we benefit from training with multiple references?

We therefore first investigate several different ways of utilizing existing human-annotated references, which include Sample One (Karpathy and Fei-Fei, 2015), Uniform, and Shuffle methods (explained in Sec. 2). Although Sample One has been explored in image captioning, to the best of our knowledge, this is the first time that an MT system is trained with multiple references.

Actually, four or five references still cover only a tiny fraction of the exponentially large space of potential references (Dreyer and Marcu, 2012). More importantly, encouraged by the success of training with multiple human references, we further propose a framework to generate many more pseudo-references automatically. In particular, we design a neural multiple-sequence alignment algo-

rithm to compress all existing human references into a lattice by merging similar words across different references (see examples in Fig. 1); this can be viewed as a modern, neural version of paraphrasing with multiple-sequence alignment (Barzilay and Lee, 2003, 2002). We can then generate theoretically exponentially more references from the lattice.

We make the following main contributions:

- Firstly, we investigate three different methods for multi-reference training on both MT and image captioning tasks (Section 2).
- Secondly, we propose a novel neural network-based multiple sequence alignment model to compress the existing references into lattices. By traversing these lattices, we generate exponentially many new pseudo-references (Section 3).
- We report substantial improvements over strong baselines in both MT (+1.5 BLEU) and image captioning (+3.1 BLEU / +11.7 CIDEr) by training on the newly generated pseudo-references (Section 4).

## 2 Using Multiple References

In order to make the multiple reference training easy to adapt to any frameworks, we do not change anything from the existing models itself. Our multiple reference training is achieved by converting a multiple reference dataset to a single reference dataset without losing any information.

Considering a multiple reference dataset  $D$ , where the  $i^{th}$  training example,  $(\mathbf{x}_i, Y_i)$ , includes one source input  $\mathbf{x}_i$ , which is a source sentence in MT or image vector in image captioning, and a reference set  $Y_i = \{\mathbf{y}_i^1, \mathbf{y}_i^2, \dots, \mathbf{y}_i^K\}$  of  $K$  references. We have the following methods to convert the multiple reference dataset to a single reference dataset  $D'$  (note that the following  $D'_{\text{sample one}}$ ,  $D'_{\text{uniform}}$  and  $D'_{\text{shuffle}}$  are ordered sets):

**Sample One:** The most straightforward way is to use a different reference in different epochs during training to explore the variances between references. For each example, we randomly pick one of the  $K$  references in each training epoch (note that the random function will be used in each epoch). This method is commonly used in existing image captioning literatures, such as (Karpathy and Fei-Fei, 2015), but never used in MT. This approach

can be formalized as:

$$D'_{\text{sample one}} = \bigcup_{i=1}^{|D|} \{(\mathbf{x}_i, \mathbf{y}_i^{k_i})\}, k_i = \text{rand}(1, \dots, K)$$

**Uniform:** Although all references are accessible by using Sample One, it is not guaranteed that all references are used during training. So we introduce **Uniform** which basically copies  $\mathbf{x}_i$  training example  $K$  times and each time with a different reference. This approach can be formalized as:

$$D'_{\text{uniform}} = \bigcup_{i=1}^{|D|} \bigcup_{k=1}^K \{(\mathbf{x}_i, \mathbf{y}_i^k)\}$$

**Shuffle** is based on Uniform, but shuffles all the source and reference pairs in random order before each epoch. So, formally it is:

$$D'_{\text{shuffle}} = \text{Shuffle}(D'_{\text{uniform}})$$

Sample One is supervised by different training signals in different epochs while both Uniform and Shuffle include all the references at one time. Note that we use mini-batch during training. When we set the batch size equal to the entire training set size in both Uniform and Shuffle, they become equivalent.

## 3 Pseudo-References Generation

In text generation tasks, the given multiple references are only a small portion in the whole space of potential references. To cover a larger number of references during training, we want to generate more pseudo-references which is similar to existing ones.

Our basic idea is to compress different references  $\mathbf{y}_0, \mathbf{y}_1, \dots, \mathbf{y}_K$  into a lattice. We achieve this by merging similar words in the references. Finally, we generate more pseudo-references by simply traversing the compressed lattice and select those with high quality according to its BLEU score.

Take the following three references from the NIST Chinese-to-English machine translation dataset as an example:

1. Indonesia reiterated its opposition to foreign military presence
2. Indonesia repeats its opposition against station of foreign troops in Indonesia
3. Indonesia reiterates opposition to garrisoning foreign armies

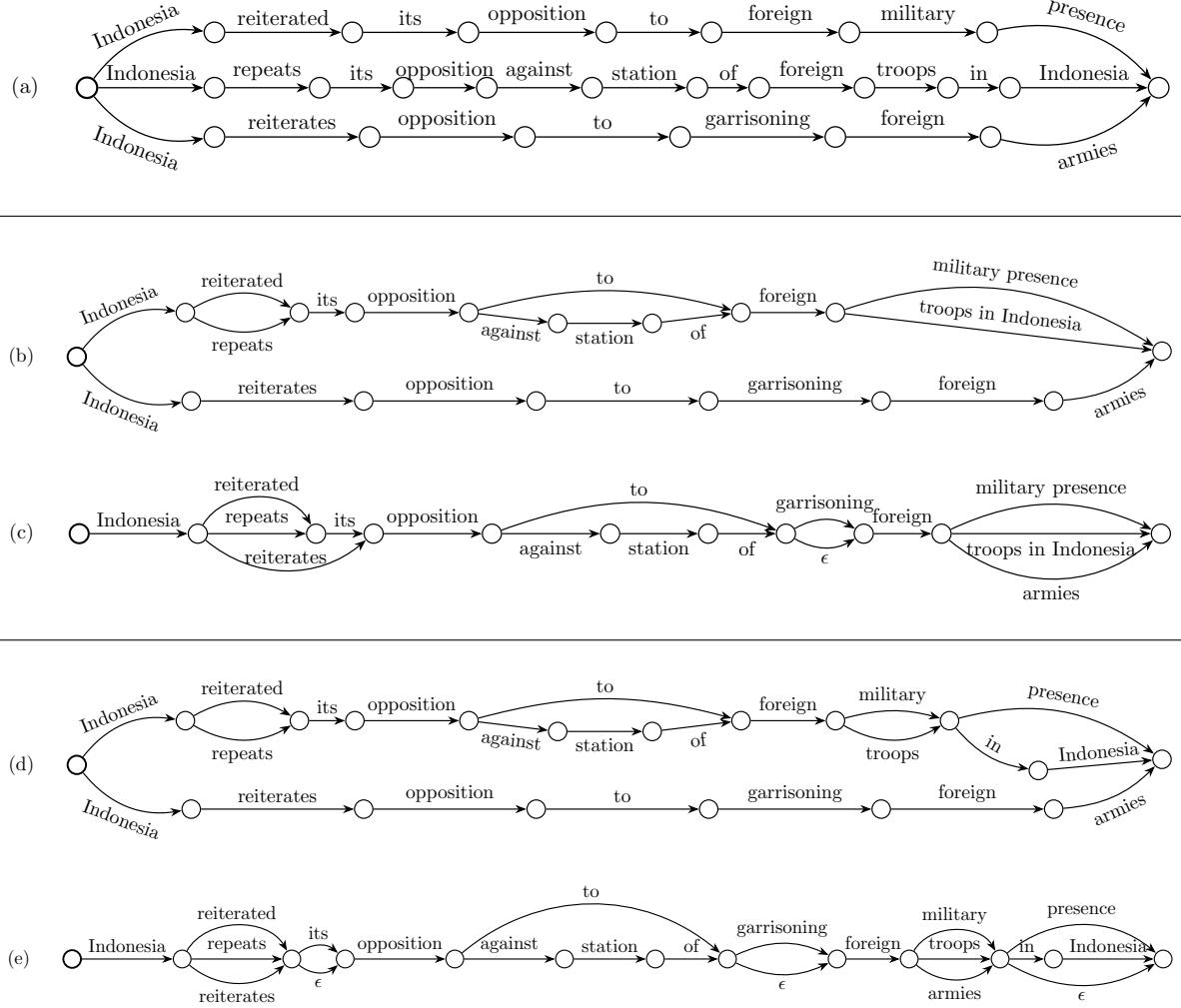


Figure 1: Lattice construction with word alignment. (b-c) is hard word alignment and 33 pseudo-references can be generated. (d-e) is soft word alignment, 213 pseudo-references can be generated.

### 3.1 Naive Idea: Hard Word Alignment

The simplest way to compress different references into a lattice is to do pairwise reference compression iteratively. At each time, we select two references and merge the same words in them.

Considering the previous example, we can derive an initial lattice from the three references as shown in Fig. 1(a). Assume that we first do a pairwise reference compression on first two references, we can merge at four sharing words: *Indonesia*, *its*, *opposition* and *foreign*, and the lattice will turn to Fig. 1(b). If we further compress the first and third references, we can merge at *Indonesia*, *opposition*, *to* and *foreign*, which gives the lattice Fig. 1(c). By simply traversing the final lattice, 33 new pseudo-references can be generated. For example:

1. *Indonesia reiterated its opposition*

*to garrisoning foreign armies*

2. *Indonesia repeats its opposition to foreign military presence*
3. *Indonesia reiterates opposition to foreign troops in Indonesia*
4. *...*

However, this simple hard alignment method (only identical words can be aligned) suffers from two problems:

1. Different words may have similar meanings and need to be merged together. For example, in the previous example, *reiterated*, *repeats* and *reiterates* should be merged together. Similarly, *military*, *troops* and *armies* also have similar meanings. If the

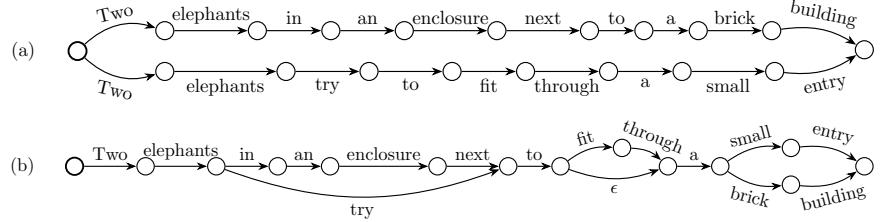


Figure 2: Mistakes from hard word alignment by merging at “to”.

lattice can align these words, we can generate the lattice shown in Fig. 1(e) which can generate 213 pseudo-references.

2. Identical words may have different meaning in different contexts and should not be merged. Considering the following two references from the COCO image captioning dataset (corresponding picture is shown in Fig. 2):

1. Two elephants in an enclosure next to a brick building
2. Two elephants try to fit through a small entry

Following the previously described algorithm, we can merge the two references at “two elephants”, at “to” and at “a”. However, “to” in the two references are very different (it is a preposition in the first reference and an infinitive in the second) and should not be merged. Thus, the lattice in Fig. 2(b) will generate the following wrong pseudo-references:

1. Two elephants try to a small entry
2. Two elephants in an enclosure next to fit through a brick building

Therefore, we need to investigate a better method to compress the lattice.

### 3.2 Measuring Word Similarity in Context

To tackle the above listed two problems of hard alignment, we need to identify synonyms and words with similar meanings. Barzilay and Lee (2002) utilize an external synonyms dictionary to get the similarity score between words. However, this method ignores the given context of each word. For example, in Fig. 1(a), there are two *Indonesia*’s in the second path of reference. If we use a synonyms dictionary, both *Indonesia* tokens will be aligned to the *Indonesia* in the first

or third sentence with the same score. This incorrect alignment would lead to meaningless lattice.

Thus, we introduce the semantic substitution matrix which measures the semantic similarity of each word pairs in context. Formally, given a sentence pair  $y_i$  and  $y_j$ , we build a semantic substitution matrix  $M = \mathbb{R}^{|y_i| \times |y_j|}$ , whose cell  $M_{u,v}$  represents the similarity score between word  $y_{i,u}$  and word  $y_{j,v}$ .

We propose a new neural network-based multiple sequence alignment algorithm to take context into consideration. We first build a language model (LM) to obtain the semantic representation of each word, then these word representations are used to construct the semantic substitution matrix between sentences.

Fig. 3 shows the architecture of the bidirectional LM (Mousa and Schuller, 2017). The optimization goal of our LM is to minimize the  $i^{th}$  word’s prediction error given the surrounding word’s hidden state:

$$p(w_i \mid \overrightarrow{\mathbf{h}}_{i-1} \oplus \overleftarrow{\mathbf{h}}_{i+1}) \quad (1)$$

For any new given sentences, we concatenate both forward and backward hidden states to represent each word  $y_{i,u}$  in a sentence  $y_i$ . We then calculate the normalized cosine similarity score of word  $y_{i,u}$  and  $y_{j,v}$  as:

$$M_{u,v} = \text{cosine}(\overrightarrow{\mathbf{h}}_u \oplus \overleftarrow{\mathbf{h}}_u, \overrightarrow{\mathbf{h}}_v \oplus \overleftarrow{\mathbf{h}}_v) \quad (2)$$

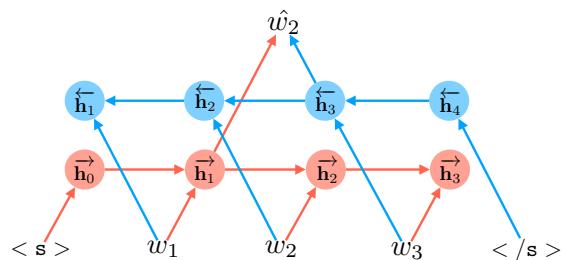


Figure 3: Bidirectional Language Model

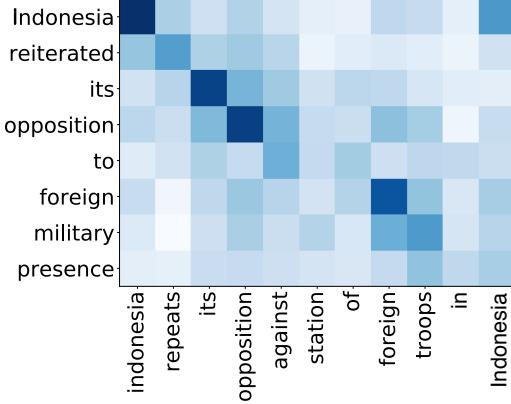


Figure 4: Semantic Substitution Matrix

Fig. 4 shows an example of the semantic substitution matrix of first two sentences in example references of Fig. 1(a).

### 3.3 Iterative Pairwise Word Alignment using Dynamic Programming

With the help of semantic substitution matrix  $M_{u,v}$  which measures pairwise word similarity, we need to find the optimal word alignment to compress references into a lattice.

Unfortunately, this computation is exponential in the number of sequences. Thus, we use iterative pairwise alignment which greedily merges sentence pairs (Durbin et al., 1998).

Based on pairwise substitution matrix we can define an optimal pairwise sequence alignment as an optimal path from  $M_{0,0}$  to  $M_{|\mathbf{y}_i|,|\mathbf{y}_j|}$ . This is a dynamic programming problem with the state transition function described in Equation (3). Fig. 5 shows the optimal path according to the semantic substitution matrix in Fig. 4. There is a gap if the continuous step goes vertical or horizontal, and an alignment if it goes diagonal.

$$opt(u, v) = \begin{cases} opt(u-1, v-1) + M_{u,v} \\ opt(u-1, v) \\ opt(u, v-1) \end{cases} \quad (3)$$

What order should we follow to do the iterative pairwise word alignment? Intuitively, we need to compress the most similar reference pair first, since this compression will lead to more aligned words. Following this intuition, we order reference pairs by the maximum alignment score  $opt(|\mathbf{y}_i|, |\mathbf{y}_j|)$  (i.e. the score of bottom-right cell in Fig. 5) which is the sum of all aligned words.

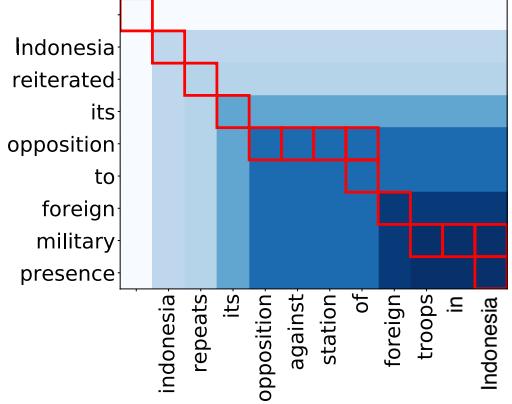


Figure 5: Dynamic Programming on Semantic Substitution Matrix

Using this order, we can iteratively merge each sentence pair in descending order, unless both the sentences have already been merged (this will prevent generating a cyclic lattice).

Since the semantic substitution matrix  $M_{u,v}$ , defined as a normalized cosine similarity, scales in  $(0, 1)$ , it's very likely for the DP algorithm to align unrelated words. To tackle this problem, we deduct a global penalty  $p$  from each cell of  $M_{u,v}$ . With the global penalty  $p$ , the DP algorithm will not align a word pair  $(\mathbf{y}_{i,u}, \mathbf{y}_{i,v})$  unless  $M_{u,v} \geq p$ .

After the pairwise references alignment, we merge those aligned words. For example, in Fig. 1, after we generate an initial lattice as shown in Fig. 1(a), we then calculate the maximum alignment score of all sentence pairs. After that, the lattice turns into Fig. 1(d) by merging the first two references (assuming they have the highest score) according to pairwise alignment shown in Fig. 5. Then we pick the sentence pair with next highest alignment score (assuming it's the last two sentences). Similar to the previous step, we find alignments according to the dynamic programming and merge to the final lattice (see Fig. 1(e)).

### 3.4 Traverse Lattice and Pseudo-References Selection by BLEU

We generate pseudo-references by simply traversing the generated lattice. For example, if we traverse the final lattice shown in Fig. 1(e), we can generate 213 pseudo-references in total.

Then, we can put those generated pseudo-references to expand the training dataset. To balance the number of generated pseudo-references for each example, we force the total number of pseudo-references from each example to be

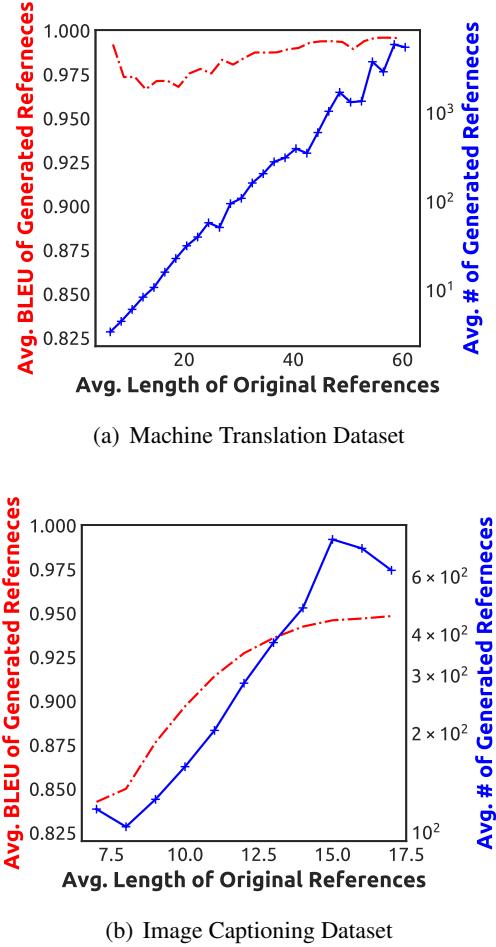


Figure 6: Analysis of generated references

$K'$ . For those examples generating  $k$  pseudo-references and  $k > K'$ , we calculate all pseudo-references' BLEU scores based on gold references, and only keep top  $K' - k$  pseudo-references with highest BLEU score.

## 4 Experiments

To investigate the empirical performances of our proposed algorithm, we conduct experiments on machine translation and image captioning.

### 4.1 Machine Translation

We evaluate our approach on NIST Chinese-to-English translation dataset which consists of 1M pairs of single reference data and 5974 pairs of 4 reference data (NIST 2002, 2003, 2004, 2005, 2006, 2008). Table 1 shows the statistics of this dataset. We first pre-train our model on a 1M pairs single reference dataset and then train on the NIST 2002, 2003, 2004, 2005. We use the NIST 2006

dataset as validation set and NIST 2008 as test sets.

Fig. 6(a) analyzes the number and quality of generated references using our proposed approach. We set the global penalty as 0.9 and only calculate the top 50 generated references for the average BLEU analysis. From the figure, we can see that when the sentence length grows, the number of generated references grows exponentially. To generate enough references for the following experiments, we set an initial global penalty as 0.9 and gradually decrease it by 0.05 until we collect no less than 100 references. We train a bidirectional language model on the pre-training dataset and training dataset with Glove (Pennington et al., 2014) word embedding size of 300 dimension, for 20 epochs to minimize the perplexity

We employ byte-pair encoding (BPE) (Sennrich et al., 2015) which reduces the source and target language vocabulary sizes to 18k and 10k. We adopt length reward (Huang et al., 2017) to find optimal sentence length. We use a two layer bidirectional LSTM as the encoder and a two layer LSTM as the decoder. We perform pre-training for 20 epochs to minimize perplexity on the 1M dataset, with a batch size of 64, word embedding size of 500, beam size of 15, learning rate of 0.1, learning rate decay of 0.5 and dropout rate of 0.3. We then train the model in 30 epochs and use the best batch size among 100, 200, 400 for each update method. These batch sizes are multiple of the number of references used in experiments, so it is guaranteed that all the references of one single example are in one batch for the Uniform method. The learning rate is set as 0.01 and learning rate decay as 0.75. We do each experiment three times and report the average result.

Table 2 shows the translation quality on the dev-set of machine translation task. Besides the original 4 references in the training set, we generate another four dataset with 10, 20, 50 and 100 references including pseudo-references using hard word alignment and soft word alignment. We compare the three update methods (Sample One, Uniform, Shuffle) with always using the first reference (First). All results of soft word alignment are better than corresponding hard word alignment results and the best result is achieved with 50 references using Uniform and soft word alignment. According to Table 3, Shuffle with original 4 references has +0.7 BLEU improvement and Uniform

Task		Pre-training	Training	Validation	Testing
Machine Translation	# of examples	1,000,000	4,667	616	691
	# of refs per example	1	4	4	4
Image Captioning	# of examples	-	113,287	5,000	5,000
	# of refs per example	-	5	5	5

Table 1: Statistics of datasets used in following experiments.

# of Refs	Method	BLEU	
		Hard Align	Soft Align
0	Pre-train	37.44	
1	First*	38.64	
	Sample One	38.81	
	Uniform	38.78	
4	Shuffle	38.87	
	Includes Pseudo-Refs		
	Sample One	37.48	39.41
10	Uniform	39.20	39.35
	Shuffle	39.13	39.53
	Sample One	37.27	38.70
20	Uniform	39.14	39.46
	Shuffle	39.12	39.42
	Sample One	37.42	37.62
50	Uniform	39.30	<b>39.65</b>
	Shuffle	38.98	39.08
	Sample One	37.54	37.63
100	Uniform	39.23	39.46
	Shuffle	38.88	39.03

Table 2: BLEU on the MT validation set. \* Baseline

# of Refs	Method	BLEU
0	Pre-train	33.58
1	First*	34.49
4	Shuffle	35.20 (+0.7)
†50	Uniform	<b>35.98</b> (+1.5)

Table 3: BLEU on the MT test set. †Includes pseudo-references generated by soft word alignment algorithm. \*Baseline.

with 50 references has +1.5 BLEU improvement. From Fig. 7(b), we can see that using the Sample One method, the translation quality drops dramatically with more than 10 references. This may be due to the higher variance of used reference in each epoch.

## 4.2 Image Captioning

For the image captioning task, we use the widely-used MSCOCO image captioning dataset. Following prior work, we use the Kapathy split (Karpathy and Fei-Fei, 2015). Table 1 shows the statistics of this dataset. We use Resnet (He et al., 2016) to extract image feature of 2048 feature size and simple fully connected layer of size 512 to an LSTM de-

# of Refs	Method	BLEU		CIDEr	
		Hard Align	Soft Align	BLEU	CIDEr
1	First	26.27		79.05	
	Sample One*	29.03		85.39	
	Uniform	30.05		89.76	
5	Shuffle	30.41		91.21	
	Includes Pseudo-Refs				
	Sample One	30.63	91.76	30.98	92.02
10	Uniform	30.40	91.48	30.77	91.89
	Shuffle	30.68	92.01	30.91	92.22
	Sample One	30.69	92.25	30.91	92.32
20	Uniform	30.73	91.69	31.03	92.61
	Shuffle	31.56	94.99	<b>31.92</b>	<b>95.59</b>
	Sample One	30.76	91.81	31.07	92.17
50	Uniform	30.66	92.30	30.99	92.61
	Shuffle	30.83	93.26	31.06	94.19

Table 4: BLEU/CIDEr on the image captioning validation set. \*Baseline.

# of Refs	Method	BLEU	CIDEr
1	First	26.70	80.70
5	Sample One*	28.67	85.41
5	Shuffle	30.94 (+2.3)	94.10 (+8.7)
†20	Shuffle	<b>31.79</b> (+3.1)	<b>97.10</b> (+11.7)

Table 5: BLEU/CIDEr on the image captioning test set with soft. † Includes pseudo-references generated by soft word alignment algorithm. \* Baseline.

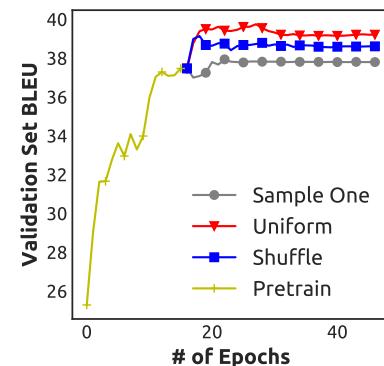
coder. We train every model for 100 epochs and calculate the BLEU score on validation set and select the best model. For every update method, we find the optimal batch size among 50, 250, 500, 1000 and we use a beam size of 5.

Fig. 6(b) analyzes the correlation between average references length with the number and quality of generated references. We set global penalty as 0.6 (which is also adopted for the generated references in the following experiments) and calculate the top 50 generated references for the average BLEU analysis. Since the length of original references is much shorter than the previous machine translation dataset, it has worse quality and fewer generated references.

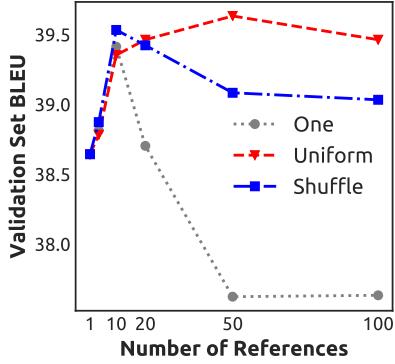
Table 4 shows that the best result is achieved with 20 references using Shuffle. This result is

Image	Original References
	a gray tabby cat is curled in a red bowl that sits on a table near a window a brown and black cat is sleeping in a bowl on a table a grey tiger cat sleeping in a brown bowl on a table an image of a cat sitting inside of a bowl on the kitchen table a cat asleep in a fruit bowl on a dining room table
Generated Lattice using Soft Alignment	
ID	Pseudo-references
1	a grey tiger cat sleeping in a brown bowl on a table near a window
2	a grey tiger cat sleeping in a brown bowl on a dining room table
3	a brown and black cat is sleeping in a bowl on the kitchen table
...	...
48	a grey tiger cat sleeping in a fruit bowl on a table
49	a cat asleep in a red bowl that sits on a table
50	a gray tabby cat is sleeping in a bowl on a table
...	...
73723	a grey and tabby cat inside of a red bowl on the dining room table
73724	a grey and tabby cat inside of a red bowl on a kitchen table
	BLEU
	100.0
	100.0
	100.0
	...
	97.1
	97.1
	97.1
	...
	0.0
	0.0

Table 6: Training example that generates maximum number of pseudo-references (73724). The selected 8 pseudo-references are sorted according to their BLEU score.



(a) Learning curve of different methods with 50 References



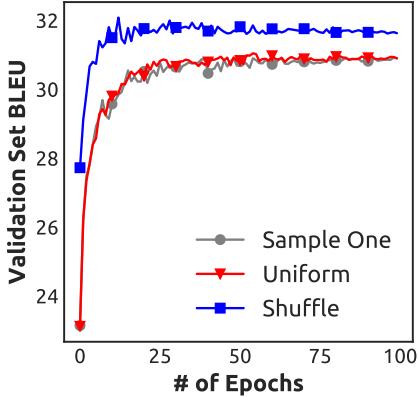
(b) MT with different number of references

Figure 7: Translation quality of machine translation task on dev-set with soft alignment

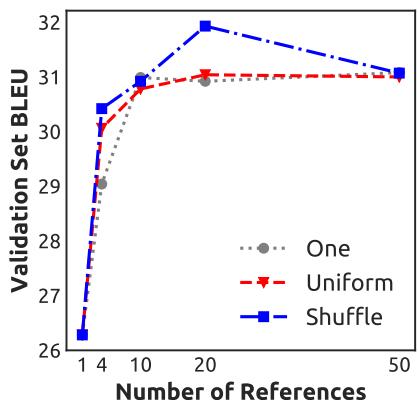
different from the result of machine translation task where Uniform method is the best. This may be because the references in image captioning dataset are much more diverse than those in machine translation dataset. Different captions of one image could even talk about different aspects. When using the Uniform method, the high variance of references in one batch may harm the model and lead to worse text generation quality. Table 5 shows that it outperforms Sample One with 4 original references, which is adopted in previous work (Karpathy and Fei-Fei, 2015), +3.1 BLEU score and +11.7 CIDEr.

### 4.3 Case Study

Fig. 6 shows a training example in the COCO dataset and its corresponding generated lattice and pseudo-references which is sorted according to its BLEU score. Our proposed algorithm generates 73724 pseudo-references in total. All the top 50 pseudo-references' BLEU scores are above 97.1 and the top three even achieve 100.0 BLEU score though they are not identical to any original references. Although the BLEU of last two sentences is 0.0, they are still valid to describe this picture.



(a) Learning curve of different methods with 20 References



(b) Image captioning with different number of references

Figure 8: Text generation quality of image captioning task on validation set with soft alignment

## 5 Conclusions

We introduce several multiple-reference training methods and a neural-based lattice compression framework, which can generate more training references based on existing ones. Our proposed framework outperforms the baseline models on both MT and image captioning tasks.

## Acknowledgments

This work was supported in part by DARPA grant N66001-17-2-4030, and NSF grants IIS-1817231 and IIS-1656051. We thank the anonymous reviewers for suggestions and Juneki Hong for proofreading.

## References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *CoRR*.

Regina Barzilay and Lillian Lee. 2002. Bootstrapping lexical choice via multiple-sequence alignment. In *Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing - Volume 10*.

Regina Barzilay and Lillian Lee. 2003. Learning to paraphrase: An unsupervised approach using multiple-sequence alignment. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*.

Markus Dreyer and Daniel Marcu. 2012. Hyter: Meaning-equivalent semantics for translation evaluation. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 162–171. Association for Computational Linguistics.

Richard Durbin, Sean Eddy, Anders Krogh, and Graeme Mitchison. 1998. Biological sequence analysis: probabilistic models of proteins and nucleic acids.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann Dauphin. 2017. Convolutional sequence to sequence learning. In *ICML*.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. *Conference on Computer Vision and Pattern Recognition CVPR*.

Liang Huang, Kai Zhao, and Mingbo Ma. 2017. When to finish? optimal beam search for neural text generation (modulo beam size). In *EMNLP 2017*.

Andrej Karpathy and Li Fei-Fei. 2015. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3128–3137.

Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. 2014. Microsoft COCO: common objects in context.

Amr Mousa and Björn Schuller. 2017. Contextual bidirectional long short-term memory recurrent neural network language models: A generative approach to sentiment analysis. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*.

Ramesh Nallapati, Bowen Zhou, and Mingbo Ma. 2016. Classify or select: Neural architectures for extractive document summarization. *CoRR*.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Alexander M. Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. *Proceedings of the 27th International Conference on Neural Information Processing Systems*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NIPS*.

Ramakrishna Vedantam, C. Lawrence Zitnick, and Devi Parikh. 2015. Cider: Consensus-based image description evaluation. In *CVPR*.

Subhashini Venugopalan, Marcus Rohrbach, Jeffrey Donahue, Raymond J. Mooney, Trevor Darrell, and Kate Saenko. 2015. Sequence to sequence - video to text. In *ICCV*.

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. 2015a. Show, attend and tell: Neural image caption generation with visual attention. *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*.

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron C. Courville, Ruslan Salakhutdinov, Richard S. Zemel, and Yoshua Bengio. 2015b. Show, attend and tell: Neural image caption generation with visual attention. In *Proceedings of Machine Learning Research*.