TRANSLATING TED SPEECHES BY RECURRENT NEURAL NETWORK BASED TRANSLATION MODEL

Youzheng Wu Xinhu Hu Chiori Hori

Spoken Language Communication Laboratory,
National Institute of Information and Communications Technology,
Kyoto, Japan
{youzheng.wu, xinhui.hu, chiori.hori}@nict.go.jp

ABSTRACT

This paper presents our recent progress on translating TED speeches[1], a collection of public lectures covering a variety of topics. Specially, we use word-to-word alignment to compose translation units of bilingual tuples and present a recurrent neural network-based translation model (RNNTM) to capture long-span context during estimating translation probabilities of bilingual tuples. However, this RNNTM has severe data sparsity problem due to large tuple vocabulary and limited training data. Therefore, a factored RNNTM, which takes bilingual tuples in addition to source and target phrases of the tuples as input features, is proposed to partially address the problem. Our experimental results on the IWSLT2012 test sets show that the proposed models significantly improve the translation quality over state-of-the-art phrase-based translation systems.

Index Terms— spoken language translation, recurrent neural network, IWSLT.

1. INTRODUCTION

The IWSLT shared task is an annual evaluation of spoken language translation organized by the International Workshop on Spoken Language Processing (IWSLT) [1]. Since 2010, the main focus of IWSLT has shifted to the translation of TED speeches, given by leaders in various fields and covering an open set of topics in technology, entertainment, design, and many others. In many TED translation systems, the phrase-based approach [2] is used, which, however, captures only bilingual context within the phrase pairs and no information outside the phrase pair is used. Therefore, they have poor generalization power.

Neural networks are experiencing significant improvements in the fields of image processing, acoustic modeling, language modeling, etc. They use continuous representation in lieu of standard discrete representation and show powerful generalization than traditional methods [3]. Recently, some pioneer studies have proposed neural network translation models to enhance generalization of translation model in statistical machine translation (SMT). The basic idea is to project the words [4] and/or phrases [5] into a continuous space and to perform the probability estimation in that space. The authors reported good improvements in the BLEU scores on some tasks. However, these translation models used feed-forward neural networks, thus, only limited context can be exploited. In language modeling, experimental results [6, 7, 8] demonstrated that recurrent neural networks (RNNs) significantly outperform feed-forward neural networks even though it is hard to train properly.

In this paper, we use word-to-word alignment to compose translation units of bilingual tuples and present a recurrent neural network-based translation model (RNNTM), which can enable the model to use arbitrary-length context theoretically. This RNNTM is expected to estimate translation probabilities of bilingual tuples more accurately. However, this RNNTM suffers poor generalization power due to large vocabulary of translation bilingual tuples. To address the problem, a factored RNNTM is proposed, which takes bilingual tuples in addition to source and target phrases of the tuples as input features. To the best of our knowledge, this is the first work to present good improvements with RNN translation models.

2. BILINGUAL TUPLE

Similar to n-gram translation model [9], we consider translation process like a language model of a particular bi-language composed of bilingual tuples that are referred to as translation units. In this way, the translation model probabilities at the sentence level are approximated by using bilingual tuples, as described by the following equation.

$$p(\mathbf{t}, \mathbf{s}) = \prod_{k=1}^{m} p(u_k | u_{k-1}, u_{k-2}, ..., u_1) \quad (1)$$

where $\mathbf{t}$ refers to target sentence, $\mathbf{s}$ to source sentence, and u_k to the k-th bilingual tuple of a given bilingual sentence pair.

[1] www.ted.com

978-1-4799-2893-4/14/$31.00 ©2014 IEEE

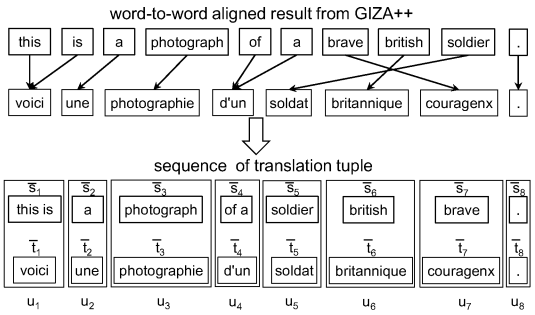

Fig. 1. Process of generating bilingual tuples from word-aligned result.

Each bilingual tuple u_k contains a source phrase s_k and its aligned target phrase t_k. Formally, u_k=s_k:t_k.

Figure 1 illustrates the process of generating translation unit u_k. Each bilingual tuple u_k is extracted from a word-to-word aligned corpus in such a way that a unique segmentation of the bilingual corpus is achieved [9]. In our implementation, GIZA++ with default settings is used to conduct word-to-word alignments in both directions, source-to-target and target-to-source [10].

Then, the main problem is to estimate the probability $p(u_k|u_{k-1}, u_{k-2}, ..., u_1)$ in Eq. 1. Prior studies have proposed many effective methods. For example, Marino et al. [9] approximated the probability by using n-gram probability $p(u_k|u_{k-1}, u_{k-2}, ..., u_{k-n+1})$ estimated via maximum likelihood and smoothing technique. Son et al. [5] and Schwenk et al [11] proposed various feed-forward neural networks to estimate the probability in continuous space. However, they are restricted to limited-length context and remain a kind of n-gram model.

3. PROPOSED METHOD

In order to use arbitrary-length context, this paper presents a recurrent neural networks-based translation model (RNNTM) to approximate the probability $p(u_i|u_{i-1}, u_{k-2}, ..., u_1)$ in Eq. 1. The RNNTM consists of an input layer, a hidden layer with recurrent connections that propagate time-delayed information, and an output layer, plus the corresponding weight matrices [12]. The input layer represents bilingual tuple encoded using 1-of-n coding, and the output layer produces a probability distribution over all tuples. The hidden layer maintains a representation of the sentence history thanks to the recurrent connections. Because the tuple u_k are bilingual pair, which results in the underlying vocabulary, hence the number of parameters, can be quite large. Table 3 in Section 4 shows the sizes of the difference vocabularies. Due to data sparsity problem, this proposed RNNTM model suffers poor generalization ability even though it is better than the smoothing approach [9].

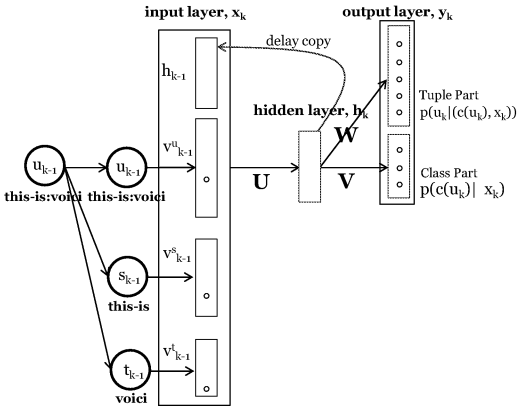

Fig. 2. Architecture of the factored RNNTM, which will go back to the RNNTM when we set $v^s_{k-1} = \mathbf{0}$ and $v^t_{k-1} = \mathbf{0}$.

3.1. Factored RNNTM

To solve the problem, we extend the RNNTM model with additional features, as shown in Figure 2. Specifically, it consists of input layer x, hidden layer h (state layer), and output layer y. The connection weights among layers are denoted by matrixes **U**, **V** and **W**. Unlike the RNNTM, which predicts probability $p(u_k|u_{k-1}, h_{k-1})$, the factored RNNTM predicts probability $p(u_k|u_{k-1}, s_{k-1}, t_{k-1}, h_{k-1})$ of generating following tuple u_k and is explicitly conditioned on the preceding tuple u_{k-1}, source of the tuple s_{k-1}, and target of the tuple t_{k-1}. It is implicitly conditioned on the entire history by the delay copy of hidden layer h_{k-1}. For convenience, u_{k-1}, s_{k-1} and t_{k-1} are called features.

In the input layer, each feature is encoded into the feature vector using the 1-of-n coding. The tuple u_{k-1}, the source phrase s_{k-1} and the target phrase t_{k-1} are encoded into $|u|$-dimension feature vector v^u_{k-1}, $|s|$-dimension feature vector v^s_{k-1} and $|t|$-dimension feature vector v^t_{k-1}, respectively. Here, $|u|$, $|s|$ and $|t|$ stand for the sizes of the tuple, the source phrase, and the target phrase vocabularies. Finally, the input layer x_k is formed by concatenating feature vectors and hidden layer h_{k-1} at the preceding time step, as shown in the following equation.

$$x_k = [v^u_{k-1},\ v^s_{k-1},\ v^t_{k-1},\ h_{k-1}] \tag{2}$$

Using this concatenation vector, the factored RNNTM can simultaneously integrate all features and the entire history in stead of backing-off to fewer features and a shorter context as factored n-gram LM does [13]. The weight of each feature is represented in connection weight matrix **U**. Therefore, it has better generalization than the RNNTM. In the special case that s_{k-1} and t_{k-1} are dropped, the factored RNNTM goes back to the RNNTM.

The hidden layer employs a sigmoid activation function:

$$h_k = f(\mathbf{U} \times x_k),\ \ f(z) = \frac{1}{1+e^{-z}}, \tag{3}$$

The output layer is split into two parts to speedup training and testing. Like [14], we map bilingual tuples into classes with frequency binning. The first part estimates the probability distribution over all classes. The second computes the probability distribution over the tuples that belong to class $c(u_k)$, the one that contains predicted tuple u_k. The computation can be expressed in Eq. 4.

$$y_k^c = g(\mathbf{V} \times h_k), \;\; y_k^t = g(\mathbf{W} \times h_k), \\ g(z_d) = \frac{e^{z_d}}{\sum_x e^{z_x}}, \tag{4}$$

Finally, probability $p(u_k|u_{k-1}, s_{k-1}, t_{k-1}, h_{k-1})$ is the product of two probability distributions.

$$p(u_k|u_{k-1}, s_{k-1}, t_{k-1}, h_{k-1}) \approx p(c(u_k)|x_i) \times p(u_k|c(u_k), x_i) \tag{5}$$

Training the RNNTM and the factored RNNTM can be performed with the back-propagation through time (BPTT) algorithm. The matrixes are randomly initialized and updated with BPTT over training data in 10-20 iterations.

4. EXPERIMENT

This paper uses the IWSLT2012 data sets, with the dev2010 as the tuning set, the tst2010, tst2011, and tst2012 as the test sets. We experiment with two language pairs, with English as source, German, French as target. For each language pair, we built a baseline phrase-based translation system using standard settings in the Moses toolkit [2] and tune it with MERT on the tuning set. The RNNTMs are used to re-score n-best lists produced by the baseline systems. The n-best size is at most 1000 for each test sentence. During the n-best re-scoring, the weights of baseline features are fixed, the weights of RNNTMs are tuned on the IWSLT dev2010 data set with the L-BFGS optimization algorithm [15]. The proposed RNNTMs are evaluated on a small task and a large task. For the parameters of both RNNTMs, we set the number of hidden neurons in the hidden layer and classes in the output layer to 480 and 300.

4.1. Small Task

In the small task, the training data only contains the speech-style bi-text, i.e., the human translation of TED speeches. Specially, the corpora for the English-French and English-German pairs contain 139K and 128K parallel sentences. The LM is a standard 4-gram language model with the Kneser-Ney discounting trained on the target side of bi-text corpus. Both of the RNNTMs are trained on the bilingual tuple sequences extracted from the same speech-style bi-text.

In Table 1, we compare the perplexities of the $n-$gram and the RNNTMs. It shows that the factored RNNTM outperforms the n-gram model by 22%.

	English-French	English-German
size of training tuples	2.01M	1.82M
n-gram	229.9	288.3
RNNTM	206.9 (10.0%)	262.3 (9.0%)
factored RNNTM	178.4 (22.6%)	222.2 (22.9%)

Table 1. Perplexities on the tst2012. The numbers in parentheses are the relative improvements over the n-gram model.

	tst2010	tst2011	tst2012
English-French			
Baseline	30.15	35.97	35.48
+RNNTM	30.51 (0.4)	36.11 (0.2)	36.44 (1.0)
+factored RNNTM	31.36 (1.2)	37.63 (1.7)	37.54 (2.1)
+Both	31.46 (1.3)	37.62 (1.7)	37.21 (1.8)
English-German			
Baseline	20.29	21.48	19.30
+RNNTM	20.67 (0.4)	21.85 (0.4)	19.56 (0.2)
+factored RNNTM	21.44 (1.2)	22.34 (0.9)	20.01 (0.7)
+Both	21.49 (1.2)	22.41 (1.0)	20.05 (0.7)

Table 2. BLEU scores for the small task. The numbers in parentheses are the absolute improvements over the baselines.

In Table 2, we summarize the results in terms of BLEU scores. The main findings are: 1. The RNNTM yields slight improvements of 0.2%-0.4% over the 1-best decoder output (Baseline) on most the test sets. 2. The factored RNNTM essentially outperforms the baseline and the RNNTM systems for all the test sets. The improvements over the baseline and the RNNTM for the English-French pair range 1.2%-2.1% and 0.8%-1.5%. For the English-German pair, the improvements over the baseline and the RNNTM are between 0.7%-1.2% and 0.5%-0.8%. This indicates that the factored RNNTM with factorization can well address the data sparsity problem of the RNNTM. For better understanding, Table 3 lists the vocabulary sizes of the tuples, source and target phrases in the RNNTMs, which shows that the vocabularies are very large. Further, the improvements for the English-German pair are comparatively smaller than that for the English-French pair. This may lie in: its vocabulary is larger and the sparsity problem is more serious. 3. Adding both of the RNNTMs, however, does not achieve significant improvements.

4.2. Large Task

In the large task, the training data includes both speech-style and text-style bi-text corpora. The text-style bi-text corpora are collected from the WMT2012 campaign (`http://www.statmt.org/wmt12`), including CommonCrawl, NewsCommentary, and Europarl. Totally, the bi-text training corpora for the English-French and English-German pairs

		Tuple	Source phrase	Target phrase
Small Task	en-fr	308K	130K	175K
	en-de	315K	148K	196K
Large Task	en-fr	1,146K	393K	605K
	en-de	1,247K	447K	715K

Table 3. Vocabulary sizes of various features.

contain 4.35M and 3.85M parallel sentences. The language model is obtained by linear interpolation of several 4-gram models trained on the target side of bi-text corpora.

The baseline systems are constructed on all the parallel corpora. However, the RNNTMs are only trained on the speech-style and the selected text-style bi-text[2]. This is because it is time-consuming to train the RNNTMs on all the available bi-text. Table 3 lists the vocabulary sizes. In Table 4, the results are reported. We observe that: 1) The proposed RNNTMs (+Both) trained on the speech-style data can even significantly enhance the baselines by 0.6%-1.2% and 0.4%-0.7% for the English-French and English-German pairs. 2) The improvements become larger with increasing of training data, for example, the Both$^{\textbf{large}}$ enhances the BLEU scores by 1.2%-1.6% for the English-French and 0.9%-1.1% for the English-German. Both of the RNNTMs can be expected to further increase the performance when we train it on bigger data.

5. RELATION TO PRIOR WORK

In SMT, neural networks are used either in language model or translation model.

In language modeling, an influential work is the feed-forward neural networks proposed by Bengio, et al. [3], in which, word is projected onto a continuous space and n-gram probabilities are estimated on that space in lieu of standard discrete space. Afterwards, Schwenk et al. [17] extended this neural network LM for statistical machine translation and improved BLEU scores significantly. Arisoy et al. [7] proposed a deep feed-forward neural network LM using multiple hidden layers instead of single hidden layer. Feed-forward neural network LMs, which predict following word based on any possible context of length n-1 history, remain a kind of n-gram LM. To address this problem, Mikolov, et al. [14] and Wu et al. [18] proposed recurrent neural network LM that can use infinite-length history theoretically.

In translation modeling, most studies consider translation process like a standard n-gram LM task by extracting tuple units from word-aligned results. Schwenk et al. [11] applied the feed-forward neural networks to estimate translation probabilities of tuple units. Le, et al. [5] improved this idea by distinguishing the source and target sides of the tuple units, to address data sparsity issues. In [4], a feed-forward neural network independent from bilingual tuples was proposed. This model can infer meaningful translation probabilities for phrase pairs not seen in the training data. However, the improvements of BLEU scores were slight. This paper is relevant to them. However, our approach uses RNN with different factorizations and can exploit long-span context. [19, 20] proposed joint language and translation modeling with RNN, in which the translation modeling slightly enhanced the BLEU scores.

[2]We employ cross entropy difference criterion [16] to select 1/8 of text-style bi-text.

	tst2010	tst2011	tst2012
English-French			
Baseline	32.92	38.67	39.41
+RNNTM	33.26 (0.4)	39.05 (0.4)	39.67 (0.3)
+factored RNNTM	33.49 (0.6)	39.77 (1.1)	40.01 (0.6)
+Both	33.50 (0.6)	39.88 (1.2)	39.95 (0.6)
+RNNTM$^{\textbf{large}}$	33.50 (0.6)	40.17 (1.5)	40.05 (0.7)
+factored RNNTM$^{\textbf{large}}$	33.41 (0.5)	39.75 (1.1)	40.19 (0.8)
+Both$^{\textbf{large}}$	34.05 (1.2)	40.31 (1.6)	40.62 (1.2)
English-German			
Baseline	22.29	23.67	20.83
+RNNTM	22.36 (0.2)	23.71 (0.0)	20.90 (0.1)
+factored RNNTM	22.86 (0.6)	24.16 (0.5)	21.44 (0.6)
+Both	22.73 (0.4)	24.16 (0.5)	21.52 (0.7)
+RNNTM$^{\textbf{large}}$	22.82 (0.5)	23.83 (0.1)	21.13 (0.3)
+factored RNNTM$^{\textbf{large}}$	23.29 (1.0)	24.64 (0.9)	22.04 (1.2)
+Both$^{\textbf{large}}$	23.19 (0.9)	24.58 (0.9)	21.92 (1.1)

Table 4. BLEU scores for the large task. The numbers in parentheses are the absolute improvements over the baselines. The RNNTMs with superscripts $^{\textbf{large}}$ means that they are trained on larger data as described in Section 4.2. The RNNTMs without superscripts means they are the models used in the small task.

6. CONCLUSION

This paper has presented recurrent neural networks (RNNs) and factored RNNs to estimate the probabilities of translation units (bilingual tuples) in a phrase-based SMT system. The experiments on the IWSLT2012 test sets show that the proposed RNNTMs can essentially improve the BLEU scores by 2.0% for the English-French pair and 1.5% for the English-German translation pair.

In the future, we will speed up the training on bigger data and evaluate them on such distant language pairs as English-Chinese (Japanese). Because the structural correspondence between English and Chinese (Japanese) is more complex than that between Indo-European language pairs.

7. REFERENCES

[1] Marcello Federico, Mauro Cettolo, Luisa Bentivogli, Michael Paul, and Sebastian Stuker, "Overview of the iwslt 2012 evaluation campaign," in *Proceedings of IWSLT 2012*, 2012.

[2] Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, and etc, "Moses: open source toolkit for statistical machine translation," in *Proceedings of ACL2007 on Interactive Poster and Demonstration Sessions*, 2007, pp. 177–180.

[3] Yoshua Bengio, Rejean Ducharme, Pascal Vincent, and Christian Jauvin, "A neural probabilistic language model," in *Journal of Machine Learning Research*, 2003, pp. 1137–1155.

[4] Holger Schwenk, "Continuous space translation models for phrase-based statistical machine translation," in *Proceedings of COLING 2012*, 2012, pp. 1071–1080.

[5] Le Hai Son, Alexandre Allauzen, and Francois Yvon, "Continuous space translation models with neural networks," in *Proceedings of HLT-NAACL 2012*, 2012, pp. 39–48.

[6] Tomas Mikolov, Anoop Deoras, Stefan Kombrink, Lukas Burget, and Jan Honza Cernocky, "Empirical evaluation and combination of advanced language modeling techniques," in *Proceedings of INTERSPEECH 2011*, 2011, pp. 605–608.

[7] Ebru Arisoy, Tara N. Sainath, Brian Kingsbury, and Bhuvana Ramabhadran, "Deep neural network language models," in *Proceedings of NAACL-HLT 2012 Workshop: Will We Ever Really Replace the N-gram Model? On the Future of Language Modeling for HLT*, 2012, pp. 20–28.

[8] Martin Sundermeyer, Ilya Oparin, Jean-Luc Gauvain, Ben Freiberg, Ralf Schlter, and Hermann Ney, "Comparison of feedforward and recurrent neural network language models," in *Proceedings of ICASSP 2013*, 2013, pp. 8430–8433.

[9] Jose B. Marino, Rafael E. Banchs, Josep M. Crego, Adria de Gispert, Patrik Lambert, Jose A.R. Fonollosa, and Marta R. Costa-jussa, "N-gram-based machine translation," in *Computational Linguistics*, 2006, vol. Volume 32 Issue 4, pp. 527–549.

[10] Franz Josef Och and Hermann Ney, "A systematic comparison of various statistical alignment models," in *Computational Linguistics*, 2003, vol. 29(1), pp. 19–51.

[11] Holger Schwenk, Marta R. Costa-jussa, and Jose A. R. Fonollosa, "Smooth bilingual n-gram translation," in *Proceedings of EMNLP/HLT 2007*, 2007, pp. 430–438.

[12] Jeffrey L. Elman, "Finding structure in time," in *Cognitive Science*, 1990, vol. 14, pp. 179–211.

[13] Kevin Duh and Katrin Kirchhoff, "Automatic learning of language model structure," in *Proceedings of COLING 2004*, 2004, pp. 148–154.

[14] Tomas Mikolov, Martin Karafiat, Lukas Burget, Jan. H. Cernocky, and Sanjeev Khudanpur, "Recurrent neural network based language model," in *Proceedings of INTERSPEECH 2010*, 2010, pp. 1045–1048.

[15] Ciyou Zhu, Richard H. Byrd, Peihuang Lu, and Jorge Nocedal, "Algorithm 778: Lbfgs-b: Fortran subroutines for large-scale bound-constrained optimization," in *ACM Transactions on Mathematical Software*, 1997, vol. 23(4), p. 550C560.

[16] Robert C. Moore and William Lewis, "Intelligent selection of language model training data," in *Proceedings of ACL 2010*, 2010, pp. 220–224.

[17] Holger Schwenk, Daniel Dchelotte, and Jean-Luc Gauvain, "Continuous space language models for statistical machine translation," in *Proceedings of COLING/ACL 2006*, 2006, pp. 723–730.

[18] Youzheng Wu, Xugang Lu, Hitoshi Yamamoto, Shigeki Matsuda, Chiori Hori, and Hideki Kashioka, "Factored language model based on recurrent neural network," in *Proceedings of COLING 2012*, 2012, pp. 2835–2850.

[19] Michael Auli, Galley Michel, Quirk Chris, and Zweig Geoffrey, "Joint language and translation modeling with recurrent neural networks," in *Proceedings of EMNLP2013*, 2013, pp. 1044–1054.

[20] Nal Kalchbrenner and Phil Blunsom, "Recurrent continuous transaltion models," in *Proceedings of EMNLP2013*, 2013, pp. 1700–1709.