Named-Entity Tagging and Domain adaptation for Better Customized Translation

Zhongwei Li[1,2], **Xuancong Wang**[1], **Ai Ti Aw**[1]
Eng Siong Chng[2], **Haizhou Li**[1,3]
[1]Human Language Technology Department, Institute for Infocomm Research (I²R), Singapore
{li-z,wangxc,aaiti}@i2r.a-star.edu.sg
[2]School of Computer Science and Engineering, Nanyang Technological University, Singapore
[3]ECE Dept, National University of Singapore, Singapore

Abstract

Customized translation need pay special attention to the target domain terminology especially the named-entities for the domain. Adding linguistic features to neural machine translation (NMT) has been shown to benefit translation in many studies. In this paper, we further demonstrate that adding named-entity (NE) feature with named-entity recognition (NER) into the source language produces better translation with NMT. Our experiments show that by just including the different NE classes and boundary tags, we can increase the BLEU score by around 1 to 2 points using the standard test sets from WMT2017. We also show that adding NE tags using NER and applying in-domain adaptation can be combined to further improve customized machine translation.

1 Introduction

As generic machine translation cannot deal well with the translation with local or specific domain context, customized machine translation is adopted to focus on the terminology of local or domain context especially for named-entities translation.

Neural machine translation (NMT) (Sutskever et al., 2014; Bahdanau et al., 2014; Luong et al., 2015) is a more recent and effective approach than the traditional statistical machine translation (SMT). It uses a large recurrent neural network (RNN) to encode a source sentence into a vector, and uses another large network to generate sentence in the target language one word at a time using the source sentence embedding and the attention mechanism.

NMT has achieved impressive result by learning the translation as an end-to-end model (Wu et al., 2016; Zhou et al. 2017; Gehring et al. 2017). Conventional NMT systems do not use linguistic features explicitly. They expect the NMT model to learn these complex sentence structures and linguistic features from big data as word embedding vectors. However, because of uneven data distribution and high linguistic complexity, there is no guarantee that NMT can capture this information and produce proper translation in all cases, especially for those terms which do not occur very often.

Recently, researchers have shown the potential benefit of explicitly encoding the linguistic features into NMT. Sennrich and Haddow (2016) proposed to include linguistic features (part-of-speech tag, lemmatized form and dependency label, morphology) at NMT source encoder side. Roee et al. (2017) instead incorporated syntactic information of target language as linearized, lexicalized constituency trees into NMT target decoder side. Their experiments showed adding linguistic information at both the source and target side can be beneficial for NMT. Based on these findings, in this paper, we propose to incorporate named-entity (NE) features to further improve neural machine translation.

Named entities play a crucial role in many monolingual and multilingual Natural Language Processing (NLP) tasks. Proper NE identification will enhance the sentence structure understanding for NMT, and thus give better translation of the named entities as well as the whole sentence.

Proceedings of the Seventh Named Entities Workshop, pages 41–46
Melbourne, Australia, July 20, 2018. ©2018 Association for Computational Linguistics

In general, named entities are more difficult to translate for NMT than SMT. This is because and NMT is weaker in translating less frequent words as compared to SMT. In addition, since there are different types of named entities, e.g. Person, Place, Organization, etc., so linguistically and logically speaking, the translation mechanisms for different types of named entities are also different. Unlike other words or phrases which occur more frequent in the training corpus, NE expressions are quite flexible, they can be composed of any character or word; moreover, in real-world applications, new named entities can emerge every day. Thus, NMT need to pay special attention to named entities to enhance the overall translation quality. Without NE context information, it is difficult to know the meaning of the words or entities with different meaning under ambiguous situation (我 喜欢 秋月。秋月 can be interpreted as person name or natural phenomenon. 三十六行 can be interpreted as a number entity or an idiomatic expression). It is also very difficult to translate number entities under never seen or rare situation (百分之 8 千点零零七).

There are many domain-based or location-based named entities. These named entities are often rare words in the document, and generally NMT cannot produce good translation for these local contexts with local named entities. Identifying local named entities and generating their translation with local context is also a challenging task which we will address in this paper. (e.g., the English name for 张志贤 is 'Teo Chee Hean' in Singapore while it's pinyin translation is 'Zhang Zhi Xian' in China)

To address the NE translation issue, some researchers work on separate models or methods while others incorporate these separate models/methods with the main NMT models (Li et al., 2016; Wang et al., 2017). They use NER to identify and align the NE pairs at both of source and target sentences, then NE pairs are replaced with NE tags for training the model; at reference stage the NE tags at target are replaced by the separate NE translation model or bilingual NE dictionary. The disadvantages of the replacement methods include NE information loss and NE alignment errors.

To avoid the complexity and disadvantages of separate model training and integration, in this paper, we add the NE type information and boundary information directly to the source sentence by a NER tool, we hope NMT will learn and understand the sentence better with this additional NE information. NE classification based on context information is important for NMT to reduce translation error under various ambiguous situations. A named entity can consist of a single word or several words, the boundary tag feature of the named entity will inform NMT model to treat these words as a single entity during translation.

Since named entities often contain local names or domain-specific names, however, the amount of local or domain-specific training data is often small. Thus, in this paper we apply domain adaptation together with named entity features to make further improvement for local context or domain-specific translation.

2 Neural Machine Translation

Machine Translation (MT) translates text sentences from a source language to a target language. SMT systems use phrases as atomic units. It obtains phrase pairs by training on large parallel corpora. NMT is a new approach in which we train a single, large neural network to maximize the translation performance. Our baseline system is based on attention-based encoder-decoder neural network model (Cho et al., 2015).

The encoder, which is often implemented as a bidirectional recurrent network with long short-term memory units (LSTM) (Hochreiter and Schmidhuber, 1997), first reads a source sentence represented as a sequence of words $\boldsymbol{x} = (x_1, x_2, \ldots x_n)$. The encoder calculates a forward sequence of hidden states and a backward sequence of hidden states. These forward and backward hidden states are concatenated to obtain the sequence of bidirectional hidden states as $\mathbf{h} = (h_1, h_2, \ldots h_n)$.

The decoder is implemented as a conditional recurrent language model that predicts a target sequence $\boldsymbol{y} = (y_1, y_2, \ldots y_m)$ given the input sequence $\boldsymbol{x} = (x_1, x_2, \ldots x_n)$. Each word y_i is predicted based on the decoder hidden state s_i, the previous word y_{i-1}, and a context vector c_i. c_i is a time-dependent content vector that is computed as a weighted-sum of the hidden states of **h:** $c_i = \sum_j a_{i,j} h_j$. The weight $a_{i,j}$of each hidden state h_j is computed by the attention model which models the probability that y_j is aligned to x_i.

The details of the attention-based multi-layer bidirectional-LSTM encoder-decoder NMT model can be found at (Cho et al., 2015). Figure 1 shows the overall system architecture.

3 NMT with NE Features and Domain Adaptation

Our main innovation over the standard sequence-to-sequence NMT model is a very simple and straight-forward way to add NE information of the source language. Compared with NE tag replacement and alignment methods (Li et al., 2016; Wang et al., 2017), our method just insert NE tags in the source sentences, there is no information loss and NE alignment issues. Since our approach does not modify the main NMT model structure, thus, our method can be applied to any sequence-to-sequence NMT model. In our model, apart from the original words in the sentence, we generate and insert NE tags which include both the NE class and NE boundary type for each NE into the sentence, thus we present the NMT encoder with the combined sentence sequence with additional NE tags.

The NE tags can be applied to both word-based and character-based source input of any language. For Chinese-to-English translation, the Chinese input can be either a word sequence or a character sequence, the English side is still word-based tokens. We segment all the unknown words as a sequence of subword units using the byte-pair encoding (Sennrich et al., 2016b).

3.1 Named-Entity Tags

For every NE in the source sentence we generate the NE class tags using the third-party tool, Stanford NER (Jenny et al., 2005):

- NE class for NE (PERSON, ORG, GPE, MISC, etc)[1]
- NE class and boundary tags: <PERSON> </PERSON> [2]

We add these NE tags to the corresponding NE of the source sentence, so as to produce the combined sentence sequence with additional NE tags.

When the source language is English, we apply subword split (@@ is the subword connector) for the out of vocabulary (OOV) words after tokenization:

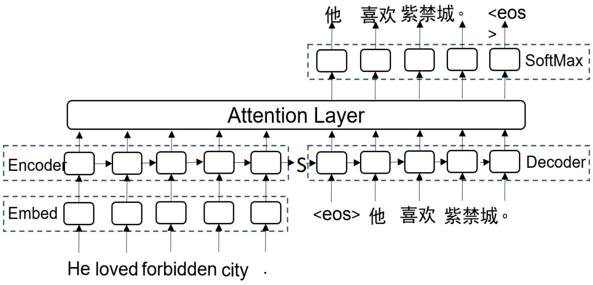

Figure 1: System Architecture

Original Source:
Patrick Roy resigns as Avalanche coach

Words and subwords with NE tags[3]*:*
<PERSON> Patrick Roy </PERSON> resigns <ORG> Avalan @@che </ORG> coach

When the source language is Chinese, we can use either word-based input or character-based input. To generate character-based input sequence for the Chinese sentence, we just split all Chinese word tokens into character tokens (English tokens are not split).

Original Source:
凯发集团成功进军中国

Words with NE tags:
<ORG> 凯发 集团 </ORG> 成功 进军 中国

Characters with NE tags:
<ORG> 凯 发 集 团 </ORG> 成 功 进 军 中 国

3.2 Preprocessing Pipeline

We design and develop the preprocessing pipeline to augment the source sequences with NE tags. It is applied on all the training set, the development set, and the test set. The preprocessing pipeline can also be used for the online translation system. The workflow of the pipeline is shown in Figure 2

The preprocessing pipeline includes the following modules:

Tokenizer: The input sentence is tokenized as word tokens.

NE Tagger: the NE tagger identifies the named entities in the sentence, and assigns the NE classes.

Subword/Chracter Splitter: We split the OOV words as subword units using byte-pair encoding (Sennrich et al., 2016b); for the Chinese

[1] ORG: Organization Entity, GPE: Geo-Political Entity
[2] <PERSON>: Start of PERSON, </PERSON>: End of PERSON

[3] *Words and subwords with NE tags are shown in blue color*

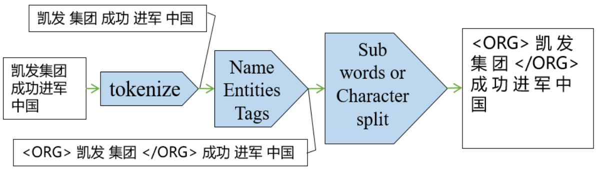

Figure 2: Preprocess Pipeline.

character-based system, we split each word as a character sequence.

Our pipeline framework is very flexible as the software components in the pipeline can be easily replaced by other software components with similar functions, for example we can, for better performance, choose different tokenizers based on the input language. For the same reason, we can switch to a different NE tagger, splitter for a different input language.

4 Experiments & Results

We have conducted our experiments with bi-direction translation between Chinese/English languages pair.

4.1 Datasets

We select the first 7 million Chinese-English sentence pairs from United Nations Parallel Corpus v1.0 (Ziemski et al., 2015), and data from LDC for the training corpus, we also select some in-domain data from local context for domain adaption training. After filtering out the long sentences (Chinese character length > 60 or number of English words > 60), the total number of sentence pairs for training is around 7 million. Table 1 shows the corpus sources for training.

Corpus	# of sentence pairs (K)	# of characters (M)
UNPCv1	6,453	1,722
LDC2017T05	63	16
LDC2017T06	6	1
LDC2006E26	35	9
In-domain	188	42
Total	6,745	1,790

Table 1: Training Data Corpus Selection.

We use the tuning sets with in-domain content for the model tuning. We use the standard test set from WMT 17 (http://www.statmt.org/wmt17/) to evaluate our model performance and compare with other models using same test set.

4.2 Data processing

We tokenize Chinese sentences using tools THULAC from Tsinghua University NLP (Zhongguo Li et al., 2009) (http://thulac.thunlp.org/), and tokenize English sentences using scripts from Moses (http://www.statmt.org/moses/). We use Stanford Named Entity Recognizer (NER) (Jenny et al., 2005) for NE Tagging for all the training, development and test data.

For character-based system, we also split every Chinese sentence as a character sequence (English words in Chinese sentences are not split into characters, but are split into subword units when OOV tokens are encountered), while the English side is still word-based. To enable open vocabulary translation, we used subword units obtained via Byte-Pair Encoding (Sennrich et al., 2016b) learning 60,000 merge operations on both Chinese and English training data.

4.3 Baseline Models

In this paper, we implement our experiment based on OpenNMT-py [4] (Klein et al., 2017) using PyTorch[5] (The PyTorch Developers, 2017). It is an open-source (MIT) neural machine translation system using Python. We train the model on one GPU: Nvidia P40. We use mini-batches of size 64, a maximum sequence length of 60, word embedding of size 600, NE boundary embedding of size 5, NE class embedding of size 10, hidden layers of size 1024, 4-layer bi-directional LSTM encoder and 4-layer uni-directional LSTM decoder. We use adam optimizer (Kingma et al., 2015) for training, we apply a dropout probability of 0.2 between LSTM stacks.

Baselines: The baseline system we trained for Chinese-to-English (ZH→EN) translation is a character-based model without any additional features, in which the Chinese source is split into characters and English is word-based with OOVs split into subword units. For ZH→EN, the performance of the character-based model is better than the word-based model. The baseline system we trained for EN→ZH translation is a word-based model, in which both source and target sentences are word tokens with OOVs split into sub-

[4] https://github.com/OpenNMT/OpenNMT-py
[5] http://pytorch.org/

word units. We found that for the baseline system without any additional linguistic features, the character-based model produces better translation than the word-based model.

Models with NE Tags: In our experiments, we train both word-based and character-based models with NE features. We found that when NE features are added, the word-based model performs better than the character-based model for both ZH→EN and EN→ZH translation.

4.4 Test Results

We calculate the performance matrix using the evaluation script *multi-bleu.perl* from Moses (Koehn et al., 2007). Two test sets are used for the evaluation; one is the standard news test set (newstest2017) from WMT 2017, while the other is our in-domain test set. Table 2 shows the performance metrics for WMT 2017 news test set for both ZH→EN and EN→ZH translation.

Models	**ZH → EN**	**EN → ZH**
Baseline	18.23	27.82
+ NE	19.92	30.38

Table 2: BLEU scores for WMT 2017 test sets

As shown in Table 2, we can see the performance improvement (around 1 BLEU score) for both directions (ZH→EN, EN→ZH) after adding NE features compared to the best baseline model.

We also apply the in-domain adaptation to the models by continue training on the in-domain data for 2-5 additional training epochs. Table 3 shows the test results for our in-domain test data.

Models	**ZH → EN**	**EN → ZH**
Baseline	14.32	21.87
+ NE	15.46	23.72
+ Adaptation	16.35	25.03

Table 3: BLEU scores for in-domain test sets

In Table 3, we show the same performance improvement when adding NE features with in-domain translation, and we also obtain further improvement for our in-domain translation by domain adaptation on top of the models with NE improvement.

5 Conclusion and Future Work

In this paper, we introduce an innovative and simple method to combine NE features and domain adaptation with NMT to improve customization translation. We add NE tags for every NE in the input sequence and pass the combined sequence to the encoder of the NMT framework. Our experiments on Chinese-to-English and English-to-Chinese translation show that adding NE features can significantly improve the performance of neural machine translation. The idea is language independent and applicable to other language pairs. Our method can also be applied to other NMT models such as the convolutional sequence-to-sequence model (Jonas Gehring et al. 2017) or the attention-only model (Vaswani et al. 2017). We also show that domain adaptation can also be applied to this method with additional improvement for in-domain text translation.

We believe that the results can be further improved by adding NE information at the target decoder side of NMT. In the future, we will explore new experiments and develop new methods to utilize the NE features to benefit translation at both source and target sides.

References

Bahdanau, Dzmitry, Cho, Kyunghyun, and Bengio, Yoshua. Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473, 2014.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander M. Rush. 2017. OpenNMT: Open-source toolkit for neural machine translation. CoRR abs/1701.02810.

Jenny Rose Finkel, Trond Grenager, and Christopher Manning.2005. Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling. *Proceedings of the 43nd Annual Meeting of the Association for Computational Linguistics* (ACL 2005), pp. 363-370.

Jie Zhou, Ying Cao, Xuguang Wang, Peng Li, and Wei Xu. 2016. Deep Recurrent Models with FastForward Connections for Neural Machine Translation. TACL 4:371–383.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. 2017. Convolutional Sequence to Sequence Learning. CoRR abs/1705.03122

Kristina Toutanova, Dan Klein, Christopher Manning, and Yoram Singer. 2003. Feature-Rich Part-of-Speech Tagging with a Cyclic Dependency Network. In Proceedings of HLT-NAACL 2003, pp. 252-259.

Kingma, Diederik P. and Jimmy Ba. 2015. “Adam: A Method for Stochastic Optimization.” The International Conference on Learning Representations. San Diego, California, USA.

Koehn, Philipp, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondˇrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. "Moses: Open Source Toolkit for Statistical Machine Translation." Proceedings of the ACL-2007 Demo and Poster Sessions, 177–180. Prague, Czech Republic.

Kyunghyun Cho, Aaron Courville, and Yoshua Bengio. 2015. Describing multimedia content using attention-based encoder-decoder networks. *IEEE Transactions on Multimedia* 17(11):1875–1886

Li, Xiaoqing; Zhang, Jiajun; Zong, Chengqing. Neural Name Translation Improves Neural Machine Translation. arXiv eprint arXiv:1607.01856, 2016.

Luong, Minh-Thang, Pham, Hieu, and Manning, Christopher D. Effective approaches to attention-based neural machine translation. In Proc. of EMNLP, 2015.

Matthew D. Zeiler. 2012. ADADELTA: An Adaptive Learning Rate Method. *CoRR*, abs/1212.5701.

Rico Sennrich and Barry Haddow. 2016a. Linguistic input features improve neural machine translation. *In Proceedings of the First Conference on Machine Translation.* Association for Computational Linguistics, Berlin, Germany, pages 83–91. http://www.aclweb.org/anthology/W16-2209.

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* (Volume 1: Long Papers). Association for Computational Linguistics, Berlin, Germany, pages 1715–1725.

Rico Sennrich, Alexandra Birch, Anna Currey, Ulrich Germann, Barry Haddow, Kenneth Heafield, Antonio Valerio Miceli Barone, and Philip Williams. 2017. The University of Edinburgh's Neural MT Systems for WMT17. In Proceedings of the Second Conference on Machine Translation, Volume 2: Shared Task Papers. Copenhagen, Denmark.

Roee Aharoni and Yoav Goldberg. 2017. Towards string-to-tree neural machine translation. *In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics.*

Sepp Hochreiter and Jurgen Schmidhuber. 1997. Long short-term memory. *Neural Comput*. 9(8):1735– 1780.

Sutskever, Ilya, Vinyals, Oriol, and Le, Quoc V. Sequence to Sequence Learning with Neural Networks. In Proc. of NIPS, pp. 3104–3112, 2014.

The PyTorch Developers. Pytorch. http://pytorch.org, 2017.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. "Attention is all you need." In Advances in Neural Information Processing Systems, pp. 6000-6010. 2017.

Wang, Yuguang, Shanbo Cheng, Liyang Jiang, Jiajun Yang, Wei Chen, Muze Li, Lin Shi, Yanfeng Wang and Hongtao Yang. "Sogou Neural Machine Translation Systems for WMT17." WMT (2017).

Wu, Yonghui, Schuster, Mike, Chen, Zhifeng, Le, Quoc V, Norouzi, Mohammad, Macherey, Wolfgang, Krikun, Maxim, Cao, Yuan, Gao, Qin, Macherey, Klaus, et al. Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation. arXiv preprint arXiv:1609.08144, 2016.

Ziemski, M., Junczys-Dowmunt, M., and Pouliquen, B. (2015). The united nations parallel corpus v1.0. In International Conference on Language Resources and Evaluation (LREC).

Zhongguo Li, Maosong Sun. Punctuation as Implicit Annotations for Chinese Word Segmentation. *Computational Linguistics,* vol. 35, no. 4, pp. 505-512, 2009.