Improved English to Russian Translation by Neural Suffix Prediction

Abstract

Neural machine translation (NMT) suffers a performance deficiency when a limited vocabulary fails to cover the source or target side adequately, which happens frequently when dealing with morphologically rich languages. To address this problem, previous work focused on adjusting translation granularity or expanding the vocabulary size. However, morphological information is relatively under-considered in NMT architectures, which may further improve translation quality. We propose a novel method, which can not only reduce data sparsity but also model morphology through a simple but effective mechanism. By predicting the stem and suffix separately during decoding, our system achieves an improvement of up to 1.98 BLEU compared with previous work on English to Russian translation. Our method is orthogonal to different NMT architectures and stably gains improvements on various domains.

Introduction

Neural machine translation (NMT) BIBREF0 has shown better performance compared with statistic machine translation BIBREF1 . Such methods encode a source sentence into hidden states and generate target words sequentially by calculating a probability distribution on the target-side vocabulary. Most NMT systems limit target side vocabulary to a fixed size, considering the limit of graphics memory size and high computing complexity when predicting a word over the whole target side vocabulary (e.g., 30K or 50K). In addition, a larger target-side vocabulary can also make the prediction task more difficult. Word-level NMT systems suffer the problem of out of vocabulary (OOV) words, particularly for morphologically rich languages. For example, English to Russian machine translation faces a big challenge due to rich morphology of Russian words, which leads to much more OOV words than some other languages. Typically a specific tag is used to represent all OOV words, which is then translated

during a post process BIBREF2 . This can be harmful to the translation quality.

There has been several methods to address this problem. Some focused on translation granularity (BIBREF3 , BIBREF3 ; BIBREF4 , BIBREF4 ; BIBREF5 , BIBREF5), while others (BIBREF6 , BIBREF6 ; BIBREF7 , BIBREF7) effectively expand target side vocabulary. However, though those methods can avoid OOV, none of them has explicitly modeled the target side morphology. When dealing with language pairs such as English-Russian, the number of different target side words is large due to the rich suffixes in Russian. The above methods are limited in distinguishing one suffix from another.

Since the total number of different stems in a morphologically rich language is much less than the number of words, a natural perspective to make a better translation on a morphologically-rich target-side language is to model stems and suffixes separately. We design a simple method, which takes a two-step approach for the decoder. In particular, stem is first generated at each decoding step, before suffix is predicted. Two types of target side sequences are used during training, namely stem sequence and suffix sequence, which are extracted from the original target side word sequence, as shown in Figure FIGREF1 . Sparsity is relieved since the number of stem types is much smaller than word types, and suffix types can be as small as several hundreds. Another advantage of this structure is that during the prediction of suffix, the previously generated stem sequence can be considered, which can further improve the accuracy of suffix prediction.

We empirically study this method and compare it with previous work on reducing OOV rates (BIBREF3 , BIBREF3 ; BIBREF4 , BIBREF4). Results show that our method gives significant improvement on the English to Russian translation task on two different domains and two popular NMT architectures. We also verify our method on training data consisting of 50M bilingual sentences, which proves that this method works effectively on large-scale corpora.

Translation Granularity

Subword based BIBREF3 and character-based (BIBREF4 , BIBREF4 ; BIBREF5 , BIBREF5) NMT are the two directions of adjusting translation granularity, which can be helpful to our problem.

In BIBREF3 (BIBREF3)'s work, commonly appearing words remain unchanged, while others are segmented into several subword units, which are from a fixed set. Both source and target side sentences can be changed into subword sequences. More specifically, some rare words are split into and represent as some more frequent units, base on a data compression technique, namely Byte Pair Encoding (BPE). The vocabulary built on common words and these frequent subword units can successfully improve the coverage of training data. In fact, a fixed size vocabulary can cover all the training data as long as the granularity of subword units is small enough. The main limitation of this method is the absence of morphology boundary. Some subword units may not be a word suffix which can represent a morphological meaning, and the subword units are treated in the same way as complete words. Subword units and complete words are predicted during a same sequence generation procedure. This may lead to two problems:

The sequence length can increase, especially on a morphologically rich language, which can lead to low NMT performance.

A subword unit cannot represent a linguistic unit, and suffix is not modeled explicitly.

BIBREF5 (BIBREF5) proposed a hybrid architecture to deal with the OOV words in source side and any generated unknown tag in the target side. In their system, any OOV words on the source side are encoded at the character level, and if an unknown tag is predicted during decoding, another LSTM will be used to generate a sequence of target-side characters, which will be used as the replacement of the

target side unknown word for the translation of a source OOV. However, their model may not work well when the target side is morphologically rich and the source side is not, because their hybrid network on the target side will only be used when an unknown tag is generated, which is always corresponding to a source unknown word. If most of the source side tokens are covered by the source vocabulary, the hybrid network may not have advantage on a morphologically rich target side language.

In BIBREF4 (BIBREF4)'s work, source side and target side sequence are all character-based, which eliminates OOV on the source side, and can generate any target side word theoretically. Character-based NMT may potentially improve the translation accuracy of morphologically rich language on the source side, but the training and decoding latency increase linearly with the sequence length, which is several times to the original word based NMT. Another disadvantage of character-based NMT is that character embedding lost the ability to represent a linguistic unit. Long-distance dependences are more difficult to be modeled in a character-based NMT. BIBREF4 (BIBREF4) use convolutional and pooling layers on the source side to make the source sequence shorter. However, the target side sequence remains much longer than the original word sequence, and suffix boundary of the target side is not specifically considered in their model. This work may more helpful if a morphologically rich language is on the source side, but it is not designed to overcome the problem brought by a morphologically rich target side language.

There is another way which can effectively reduce target-side OOV. Both BIBREF6 (BIBREF6) and BIBREF7 (BIBREF7) use a large target-side vocabulary. To overcome the problem of GPU memory limitation and increasing computational complexity, instead of the original vocabulary, a selected subset is actually used both during the training and decoding time. Their model can generate any of the words in the large vocabulary, but data sparsity still remains, the low frequent words in the training data is not fully trained.

Morphology and MT

Previous work considered morphological information for both SMT and NMT. BIBREF8 (BIBREF8) proposed an effective way to integrate word-level annotation in SMT, which can be morphological, syntactic, or semantic. Morphological information can be utilized not only on source side, but also the target side. Although these annotation can help to improve the translation procedure, data sparsity still exists. BIBREF9 (BIBREF9) decompose the process of translating a word into two steps. Firstly a stem is produced, then a feature-rich discriminative model selects an appropriate inflection for the stem. Target-side morphological features and source-side context features are utilized in their inflection prediction model.

BIBREF10 (BIBREF10) use distributed representations for words and soft morphological tags in their neural inflection model, which can effectively reduce lexical sparsity, leading to less morphological ambiguity. This is the first try of modeling inflection through a neural method, integrated in a SMT architecture.

For NMT, BIBREF11 (BIBREF11) make use of various source side features (such as morphological features, part-of-speech tags, and syntactic dependency labels) to enhance encoding in NMT. This is the first time morphological information is leveraged in NMT architecture. Target-side morphology is not considered in their work. BIBREF12 (BIBREF12) predict a sequence of interleaving morphological tags and lemmas, followed by a morphological generator. They used a external model to synthesize words given tags and lemmas. Our method is the first to explicitly consider the generation of morphological suffixes within a neural translation model. Our work is motivated by a line of work that generates morphology during text generation (BIBREF13 , BIBREF13 ; BIBREF14 , BIBREF14 ; BIBREF10 , BIBREF10).

Russian Morphology and Stemming

Morphology Russian has rich morphology, which includes number (singular or plural), case (nominative, accusative etc.), gender (feminine, masculine or neuter) and tense mood. Figure FIGREF2 shows one example for Russian. A noun word “ball” is always masculine, but the suffix differs when the case and number changes, resulting in 10 different forms. Some other nouns can be feminine or neuter, and their adjectives will agree with them. Both adjectives and verbs have different forms according to their case, tense mood and the form of words they modify. Such morphological changes bring a challenge to machine translation task.

Stemming A Russian word can be split into two parts, namely the stem and the suffix. Suffix contains morphological information of a Russian word, including gender, number and case etc. In this paper, we use a deterministic rule-based stemmer to obtain stem and suffix for a Russian word. The process of stemming is shown in Figure FIGREF1 .

Neural Machine Translation Baselines

We experiment with two different types of Neural Machine Translation (NMT) systems, one using a recurrent encoder-decoder structure BIBREF0 , the other leveraging the attention mechanism on the encoder BIBREF15 .

Recurrent Neural Network Based NMT We use an encoder-decoder network proposed by BIBREF16 (BIBREF16). The encoder uses a bi-directional recurrent neural network (RNN) to encode the source sentence, the decoder uses a uni-directional RNN to predict the target translation. Formally, the source sentence can be expressed as INLINEFORM0 , where INLINEFORM1 is the length of the sentence. It is encoded into a sequence of hidden states INLINEFORM2 , each INLINEFORM3 is the result of a concat

operation on a forward (left-to-right) hidden state INLINEFORM4 and a backword (right-to-left) hidden state INLINEFORM5 : DISPLAYFORM0 DISPLAYFORM1

INLINEFORM0 is a variation of LSTM BIBREF17 , namely Gated Recurrent Unit (GRU) BIBREF18 : DISPLAYFORM0 DISPLAYFORM1

where INLINEFORM0 , INLINEFORM1 , INLINEFORM2 are weight matrices which are learned.

During decoding, at each time step INLINEFORM0 , an attention probability INLINEFORM1 to the source word INLINEFORM2 is first calculated by: DISPLAYFORM0

and DISPLAYFORM0

is an attention model that gives a probability distribution on source words INLINEFORM0 , which indicates how much the source word INLINEFORM1 is considered during the decoding step INLINEFORM2 to generate target side word INLINEFORM3 . The attention layer INLINEFORM4 can be as simple as a feed-forward network. INLINEFORM5 is a weighted sum of the encoding hidden state at each position of input sentence: DISPLAYFORM0

INLINEFORM0 is then fed into a feed-forward network together with previous target word embedding INLINEFORM1 and the current decoding hidden state INLINEFORM2 to generate the output intermediate state INLINEFORM3 : DISPLAYFORM0

and DISPLAYFORM0

where INLINEFORM0 is GRU, which is mentioned before. The output intermediate state INLINEFORM1

is then used to predict the current target word by generating a probability distribution on target side vocabulary. In our implementation, maxout BIBREF19 mechanism is used in both training and decoding. Dropout BIBREF20 is used in training time.

Transformer BIBREF15 is a recently proposed model for sequence to sequence tasks. It discards the RNN structure for building the encoder and decoder blocks. Instead, only the attention mechanism is used to calculate the source and target hidden states.

The encoder is composed of stacked neural layers. In particularly, for the time step INLINEFORM0 in layer INLINEFORM1 , the hidden state INLINEFORM2 is calculated as follows: First, a self-attention sub-layer is employed to encode the context. For this end, the hidden states in the previous layer are projected into a tuple of queries(INLINEFORM3), keys(INLINEFORM4) and values(INLINEFORM5), where INLINEFORM6 in the following function denotes a feed forward layer: DISPLAYFORM0

Then attention weights are computed as scaled dot product between current query and all keys, normalized with a softmax function. After that, the context vector is represented as weighted sum of the values projected from hidden states in the previous layer. The hidden state in the previous layer and the context vector are then connected by residual connection, followed by a layer normalization function BIBREF21 , to produce a candidate hidden state INLINEFORM0 . Finally, another sub-layer including a feed forward layer, followed by another residual connection and layer normalization, are used to obtain the hidden state INLINEFORM1 : DISPLAYFORM0

The decoder is also composed of stacked layers. The hidden states are calculated in a similar way, except for the following two differences: First, only those target positions before the current one are used to calculate the target side self-attention. Second, attention is applied in both target-to-target and target-to-source. The target-to-source attention sub-layer is inserted between the target self-attention

sub-layer and the feed-forward sub-layer. Different from self-attention, the queries(INLINEFORM0) are projected from target hidden states in the previous layer, and the keys(INLINEFORM1) and values(INLINEFORM2) are projected from the source hidden states in the last layer.

The rest of the calculation is exactly the same with self-attention. Compared to RNN based sequence to sequence models, transformer allows significantly more parallelization, since all the hidden states in the same layer can be calculated simultaneously, whereas the hidden states in RNN can only be calculated sequentially from left to right. In consideration of translation quality, BIBREF15 (BIBREF15) use multi-head attention instead of single-head attention as mentioned above, and positional encoding is also used to compensate the missing of position information in this model.

Target-Side Suffix Prediction

We take a two-step approach for the decoder, yielding a stem at each time step before predicting the suffix of the stem. Since we only make use of source hidden states, target hidden states, target to source attention weights and target predicted tokens, these are universal in all sequence to sequence models, our method can be implemented into any of these models.

Figure FIGREF23 shows a more detailed procedure. Decoding target stems is exactly the same as decoding target words in normal sequence to sequence model, which is predicted through a softmax layer based on the target output layer. All we need is to replace target words with target stems:
DISPLAYFORM0

where INLINEFORM0 is a weight matrix to transfer the output layer INLINEFORM1 from a dimension of hidden size to target side vocabulary size. INLINEFORM2 is target side hidden state at time step INLINEFORM3 when generating the stem. INLINEFORM4 is the output state: DISPLAYFORM0

INLINEFORM0 is a single layer feed-forward neural network.

After the prediction of INLINEFORM0 , the target suffix INLINEFORM1 on decoding step INLINEFORM2 is immediately predicted from the target suffix hidden state INLINEFORM3 : DISPLAYFORM0

INLINEFORM0 is generated from a single layer feed-forward neural network by using the stem embedding INLINEFORM1 , stem hidden state INLINEFORM2 , and source context vector INLINEFORM3 : DISPLAYFORM0

Since we consider that the attention degree towards each word in the source sequence is useful to the generation of suffix, the aligned source context is also used during the prediction of suffix. Note that the source context vector INLINEFORM0 is shared between the generation of stem hidden state INLINEFORM1 and suffix hidden state INLINEFORM2 .

In addition, the embedding of the predicted suffix is not further fed into the hidden state of the next stem, because we think suffix information can provide little information for predicting the next stem from a linguistic perspective.

Training

During the training stage, the objective function INLINEFORM0 consists of two components: DISPLAYFORM0

where: DISPLAYFORM0

and DISPLAYFORM0

INLINEFORM0 verifies from 0 to 1, and INLINEFORM1 can also be modeled in the whole architecture, which will be studied in our future work. In our experiments, we set INLINEFORM2 to 0.1 empirically. We use Adam BIBREF22 as our optimizing function.

Decoding

Beam search is adopted as our decoding algorithm. At each time step, the search space can be infeasible large if we take all the combinations of stems and suffixes into consideration. So we use cube pruning BIBREF23 to obtain n-best candidates. First, the top INLINEFORM0 stems with the highest scores are pushed to the stack. Then for each stem, we predict the top INLINEFORM1 suffixes, which will result in INLINEFORM2 complete candidates. The candidates will be inserted to a priority queue, which keeps records of the top INLINEFORM3 complete candidates. After all the stems are expanded, the final n-best candidates are obtained.

Experiments

We run our experiments on English to Russian (En-RU) data under two significantly different domain, namely the news domain and the e-commerce domain. We verify our method on both RNN based NMT architecture and Transformer based NMT architecture.

Data

News We select 5.3M sentences from the bilingual training corpus released by WMT2017 shared task on the news translation domain as our training data. We use 3 test set, which are published by WMT2017 news translation task, namely “News2014”, “News2015”, “News2016”.

E-commerce We collect 50M bilingual sentences as our training corpus:

10M sentences are crawled and automatic aligned from some international brand's English and Russian websites.

20M are back translated corpus: First we crawled the Russian sentences from websites of certain Russian's Brands. Then translated them to English through a machine translation system trained on limited RU-EN corpus BIBREF24 .

The last 20M bilingual sentences are crawled from the web, and are not domain specific.

We typically use the following 3 types of data as test set, which are named title, description and comment, these sentences are all extracted from e-commerce websites. Title are the goods' titles showed on a listing page when some buyers type in some keywords in a searching bar under an e-commerce website. Description refers to the information in a commodities' detail page. Comment include the review or feedback from some buyers. Example sentences are shown in Table TABREF33 . For each kind of test set, we randomly select 1K English sentences and translate it by human.

Pre-Processing Both the training set and the test set are lowercased, and some entity words appeared in the data are generalized into specific symbols, such as “_date_”, “_time_”, “_number_”. When selecting our training data, we keep the sentences which has length between 1 to 30. We use a bilingual sentence scorer to discard some low-quality bilingual sentences. The scorer is simply trained under algorithm of IBM Model 1 BIBREF25 on a very large bilingual corpus.

Target Side Word Stemming We use snowball to create stems from words. Because stem created from snowball is always a substring of the original word, we can obtain suffixes by simply applying a string cut

operation. By applying snowball to a target side word sequence, we split a target side sentence into a stem sequence and a suffix sequence. The stemming accuracy of snowball is 83.3% on our human labeled test set.

Applying BPE to Target Side Stem Sequence We also use the Byte-pair encoding (BPE algorithm) on the target side stem sequence, which will further reduce data sparsity. Some stems will be split into “sub-stem” units. The stem sequence is transferred to “sub-stem” sequence at this step. Suffix sequence should also be adjusted according to the “sub-stem” sequence simultaneously. More specifically, as shown in Figure FIGREF36 , if a stem is split into INLINEFORM0 “sub-stem” units, then INLINEFORM1 “N” (refers to “N” in Figure FIGREF1) will be inserted into the suffix sequence, and these tags will be located in front of the suffix which is corresponding to the original complete stem. The sub-stem sequence and the adjusted suffix sequence are the final training corpus on target side.

Baselines

Our RNN and Transformer baseline systems utilize BPE BIBREF3 to transfer the original word sequence to subword sequence on both the source and the target sides, since the subword method had a stable improvement compared with word based system, especially on morphologically rich languages.

Besides, we compared our system with a fully character-based baseline system, which is an implementation of BIBREF4 (BIBREF4)'s work, and is available on github.

We limit the source and target vocabularies to the most frequent 30K tokens for both English and Russian. For news domain, about 99.7% tokens are covered by the source side vocabulary, about 97.0% target tokens are covered by the target side vocabulary.

Our System

For our system, the source token coverage is the same as the baselines. On the other hand, 100% target tokens are covered by the target-side vocabulary, which consists of “sub-stem” units generated from target side stem sequence by applying BPE algorithm. There are totally 752 types of suffixes, which are calculated from the suffix sequences generated from target side sentences.

Distributed Training

For the experiments on the e-commerce domain, the training data is large. We use a distributed training framework for both the baseline system and our system. Training data are split into several parts, each being trained on a single worker node. A parameter server averages the model parameters from each worker node after every 100 training batchs and then synchronizes the averaged model to every worker node. Each worker continues with the training process based on the averaged model.

Results and Analysis

We use BLEU BIBREF26 as our evaluation metric. The performance of different systems are shown in Table TABREF34 and TABREF35 . On both the news and e-commerce domains, our system performs better than baseline systems.

On news domain, the average improvement of our method is 1.75 and 0.97 BLEU score when implemented on RNN-based NMT, compared with subword BIBREF3 method and fully character-based BIBREF4 method, respectively. When implemented on Transformer BIBREF15 , average improvement is 1.47 BLEU compared with subword method. On the e-commerce domain, which use 50M sentences as training corpus, the average improvement of our method is 0.68 BLEU compared with the subword

method.

We evaluate stem accuracies and suffix accuracies separately. For stem, we use BLEU as evaluation metric, Table TABREF34 shows stem BLEU of different methods on “News2014” test set, our method can gain significant improvement compared with baselines, since our method can reduce data sparsity better than baselines. Our method can effectively reduce suffix error, Figure FIGREF43 gives some examples both on e-commerce and news domains:

For the first sample, the suffix of the translation words (tagged by 1 and 2) from two different baseline systems means a reflexive verb, whose direct object is the same as its subject. In other words, a reflexive verb has the same semantic agent and patient. It is an incorrect translation according to the source meaning, because we can infer from the source sentence that the agent is a person and the patient is an object (some goods bought by a customer). In our system, the suffix of the translation word (tagged by 3) is correct. It represents an infinitive verb which may take objects, other complements and modifiers to form a verb phrase.

In the second sample, the translation word (tagged by 1) is not accurate, its suffix represents a plural form, but the correct form is singular, because the corresponding source word “positive” is singular form. Character-based system can correctly translate source word “stars” into a Russian word with plural form. However, the translation of “positive” (tagged by 2) is still with wrong form. Both the translation of “positive” and “stars” from our system are with the correct forms.

In the third sample, the translation word tagged by 3 represents past tense; However, the translation words tagged by 1 and 2 represent present tense. Our system successfully predicted the tense moods.

Conclusion

We proposed a simple but effective method to improve English-Russian NMT, for which a morphologically rich language is on the target side. We take a two-step approach in the decoder. At each step, a stem is first generated, then its suffix is generated. We empirically compared our method with two previous methods (namely subword and fully character-based), which can also to some extent address our problem. Our method gives an improvement on two encoder-decoder NMT architectures on two domains. To our knowledge, we are the first to explicitly model suffix for morphologically-rich target translation.

Acknowledgments

We thank the anonymous reviewers for their detailed and constructed comments. Yue Zhang and Min Zhang are the corresponding authors. The research work is supported by the National Natural Science Foundation of China (61525205, 61432013, 61373095). Thanks for Xiaoqing Li, Heng Yu and Zhdanova Liubov for their useful discussion.