Radical-Enhanced Sequence to Sequence Model for Chinese-English Neural Machine Translation

Kevin Tan, Xinwen Wang
UC San Diego
{kstan, xiw315}@ucsd.edu

Abstract

Current state-of-the-art neural machine translation (NMT) architectures use an encoder-decoder structure for transforming a sequence of words in one language to a sequence of words in another language. However, these models typically use a word-level approach that does not take advantage of additional language-specific semantics. Focusing on Chinese to English translation, we implement a sequence to sequence model with global attention mechanism in PyTorch using Chinese radicals to enhance the learned word vector embeddings. Specifically, we learn embedding weight matrices for word-level, character-level, and radical-level and combine them together as input to the encoder. We train our model on the casia2015 dataset and evaluate it by computing the BLEU score.

1 Introduction

In this paper, we tackle the problem of neural machine translation (NMT), which can be described as the task of converting a sequence of words in the source language to a sequence of words in the target language. In the past, translation systems were based on rule based approaches or statistical methods. However, recent advancements in deep learning and in particular, vector embeddings as word representations, have proved to be very effective in the past 3 years [1]. We focus on sequence to sequence models which operate in an encoder-decoder framework. At a high level, the encoder takes the model's input sequence and encodes it into a fixed-size hidden vector, and the decoder takes that context vector and unravels it word by word to generate an output sequence.

Specifically, we improve the performance of Chinese machine translation by using Chinese radicals as the basic semantic unit for embedding. In Chinese linguistics, an individual Chinese character can be decomposed into its radical subparts, each of which is often a semantic indicator that contributes to the overall meaning of the character [7]. We demonstrate how delving deeper into radical-level representations can achieve better results than modern methods which typically rely on word-level representations.

2 Background

Current Chinese translators process text materials at the level of individual characters, which are in turn made from smaller units called radicals. An interesting fact about Chinese radicals is their pictographic nature, which means the literal shape of a radical is indicative of its possible meanings. Thus, radicals that have similar pictographic shape are more likely to have similar meanings; in addition, they will be closer in the vector embedding space of which they are mapped into. So compared to character embeddings, radical embeddings can break down the surrounding characters and provide more meaningful semantic information.

Consider, for example, the word embedding of the context character 胃 (which means stomach) will make predictions about food, eating or other organs. However, if we use a radical embedding,

Figure 1: Chinese radicals containing additional semantic information per character

the radicals for this character are 田 and 月. 月 means things relating to meat or the way people process meat. Using radical embeddings allows us to obtain more semantic information per character, granting the model a wider range of possible target characters to predict.

3 Approach

3.1 Data Collection

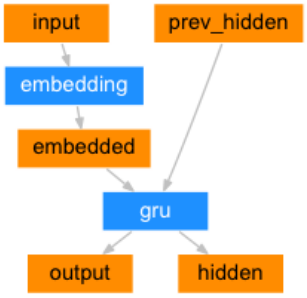
In order to train a neural machine translator, we need a parallel corpus consisting of sentences in Chinese that are aligned correspondingly with sentences in English. We explored many options and decided to use a dataset [1] shared by the China Workshop on Machine Translation (CWMT) community as part of the EMNLP 2017 Workshop on Machine Translation. The casia2015 dataset consists of one million parallel sentence pairs automatically collected from the web, in the domain of news articles (political, international, finance, forum, eduction, etc). It is provided by the Institute of Automation, Chinese Academy of Sciences.

3.2 Sequence to Sequence Model Architecture

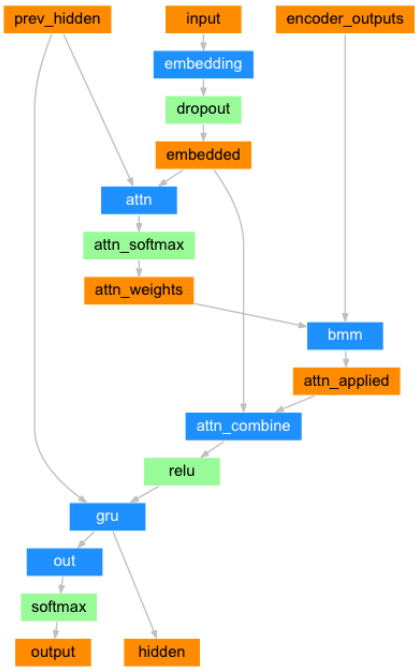
We implemented a simple sequence-to-sequence model with global attention, which utilizes two recurrent neural networks working together to transform one sequence to another. The encoder network transforms a sequence of symbols into a list of vectors with one vector per until symbol, condenses an input sentence in Chinese into a final hidden state vector. Given this hidden vector state, the decoder network unfolds it into the corresponding translated English sentence, one symbol at a time until the end of sentence (EOS) symbol is produced. The encoder and decoder module are connected via an attention network which allows the decoder to focus on different time steps of the source sentence during the course of decoding.

Formally, let X, Y be the source and target sentence pair, $X = x_1, x_2, ...x_M$ be the sequence of M Chinese words, and $Y = y_1, y_2, ...y_N$ be the sequence of N English words. The encoder repeatedly generates the hidden vectors $h = (h_1, h_2, ...h_M)$ over the source sentence. Each hidden state is

[1] http://nlp.nju.edu.cn/cwmt-wmt/



(a) Encoder architecture



(b) Decoder architecture

Figure 2: Sequence to sequence model architecture

defined as $h_j = GRU(h_{j-1}, x_j)$, where h_{j-1} is the hidden state at the previous time step and x_j is the input word to be transformed into a vector as input to the Gated Recurrent Unit (GRU) [2]. Loosely speaking, the GRU is characterized by a unique gating mechanism consisting of an update gate and reset gate that influences the output hidden state.

3.3 Attention Decoder

To implement attention mechanism [3], we take input from each time step of the encoder, but give a particular weightage to the timesteps. The weightage depends on the importance of that time step for the decoder to optimally generate the next word in the sequence, as shown in Figure 1b. The basic idea behind attention is that instead of attempting to learn a single vector representation for each sentence, we instead maintain vectors for each word in the input sentence, and reference these vectors at each decoding step.

Formally, the decoder is also an RNN that predicts the next word y_t given the context vector c_t, the hidden state of the decoder s_t and the previous predicted word y_{t-1}, which is computed by:

$$p(y_t|y_{<t}, x) = softmax(GRU(s_t, y_{t-1}, c_t)) \tag{1}$$

The attention context vectors c_t can be obtained and calculated as follows:

$$s_t = GRU(s_{t-1}, h_j) \tag{2}$$

$$\alpha_j = \frac{exp(s_t)}{\sum_{k=1}^{M} exp(s_t)} \tag{3}$$

$$c_t = \sum_{j=1}^{M} \alpha_j h_j \tag{4}$$

3.4 Chinese Word Segmentation and Radical Lookup

In Chinese, words can be formed from multiple characters, so it is difficult to tokenize a sentence into its word subparts without a basic understanding of the meanings of the words. To overcome

this, we leverage a Chinese Word Segmenter that does not rely on a large corpus. At a high level, the segmenter works by using information entropy of measure how random the set of left and right words of a text fragment are. Potential words are treated as all the possible substrings that do not exceed an upper bound max length limit, and only the words which reach a minimum entropy threshold are extracted. In our experiments, we find that max word length of 3 and minimum entropy of 0.5 work best for our results. [2]

3.5 Radical-Enhanced Embedding Strategy

Given an input sentence in Chinese, we need to form the radical-enhanced vector embedding for each word. To do this, we first parse the sentence into word sub-components by the strategy explained in Section 3.4. We simultaneously learn vector embeddings for word-level, character-level, and radical-level. Each word w_j can be split into characters $c_j = (c_{j1}, c_{j2}, ...c_{jm})$ which are obtained by tokenizing on white space. Then, each character c_j can be further split into radicals $r_j = (r_{j1}, r_{j2}, ...r_{jn})$ which are obtained by the Xinhua Zidian radical lookup table [3]. Then, the net character and radical representations are computed as:

$$z_j = \sum_{k=1}^{m} z_{jk} \tag{5}$$

$$r_j = \sum_{k=1}^{n} r_{jk} \tag{6}$$

and the final embedding vector is

$$x_j = [w_j; z_j; r_j] \tag{7}$$

where ':' is the concatenate operation.

4 Experiments

We train the model using stochastic gradient descent algorithm [6] with learning rate of 0.01, for a total of 75000 epochs. The encoder and decoder networks are both initialized with hidden state of 256, and dropout 0.1 for attention decoder. For the encoder, the embedding weight matrix is (942, 256), GRU weights are (768, 256) with biases (768,). For the decoder, the embedding weight matrix is (994, 256), with attention weights (150, 152) and biases (150,), combining vector weights (256, 512) and bias (256,) with GRU weights (768, 256). The final output weight layer is (994, 256).

4.1 BLEU metric

We use the Bilingual Evaluation Understudy Score (BLEU)[4] as a metric for evaluating a generated sentence to a reference sentence. A perfect match results in a score of 1.0, whereas a perfect mismatch results in a score of 0.0. We utilize a widely-used, standard BLEU score scheme implemented in the Python NLTK library. We ran our model against 1000 randomly generated pairs and compared the ground truth with the translated candidate sentence. In some cases, there are no 2-gram matches amongst the reference and candidate, so we use Smoothing Method 4 to give them proportionally smaller smoothed counts.

We obtain a BLEU score of 0.287 averaged over 1000 randomly generated sentence pairs.

5 Conclusion

We explored how Chinese radicals can enhance vector word embeddings for sequence to sequence models for the problem of Chinese to English machine translation. We trained the model on a large parallel corpus of data containing aligned pairs of Chinese and English sentences, and observed through human evaluation that the model can sensibly produce correct English sentences in terms

[2] https://github.com/Moonshile/ChineseWordSegmentation

[3] https://en.wikipedia.org/wiki/Xinhua_Zidian

of grammar, syntax, and spelling, but occasionally fails to closely translate the input sentence to match the target. We calculate quantitative metrics for measuring how close the translated candidate sentence matched the reference sentences, obtaining an average BLEU score of 0.287 over 1000 trials.

This suggests several opportunities for future work. Firstly, the study could be conducted with a more rigorous engineering effort in using a larger corpus than the small subset we trained on with additional computing resources. Secondly, it would be worthwhile to investigate more into various network architectures such as bidirectional LSTM, stackled LSTM, or different decoder prediction methods like beam search. Thirdly, it would be beneficial to visualize the vector space spanned by radical embeddings to determine if they exhibit any interesting characteristics.

Acknowledgments

We thank Professor Leon Bergen for being our mentor and all the teaching staff for their assistance with this project. We also thank our peers for the helpful feedback during the project presentations on the last day of class.

References

[1] Cho, K., van Merrienboer, B., Bahdanau, D., Bengio, Y.: On the properties of neural machine translation: Encoder-decoder approaches. CoRR abs/1409.1259 (2014), http://arxiv. org/abs/1409.1259

[2] Chung, J., Gulcehre, C., Cho, K., Bengio, Y.: Empirical evaluation of gated recurrent neural networks on sequence modeling. Presented in NIPS 2014 Deep Learning and Representation Learning Workshop (2014)

[3] Luong, M., Pham, H., Manning, C.D.: Effective approaches to attention-based neural machine translation. CoRR abs/1508.04025 (2015), http://arxiv.org/abs/1508. 04025

[4] Papineni, Kishore, Salim Roukos, Todd Ward, and WeiJing Zhu. BLEU: A Method for Automatic Evaluation of Machine Translation, Association of Computational Linguistics, 2002

[5] Shi, X., Zhai, J., Yang, X., Xie, Z., Liu, C.: Radical embedding: Delving deeper to chinese radicals. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers). pp. 594–598. Association for Computational Linguistics (2015). https://doi.org/10.3115/v1/P15-2098, http://aclanthology.coli.uni-saarland.de/pdf/P/P15/P15-2098.pdf

[6] Zeiler, M.D.: Adadelta: an adaptive learning rate method. arXiv preprint arXiv:1212.5701 (2012)

[7] Zhang, J., Matsumoto, T.: Improving character-level japanese-chinese neural machine translation with radicals as an additional input feature. In: 2017 International Conference on Asian Language Processing (IALP). pp. 172–175 (Dec 2017). https://doi.org/10.1109/IALP.2017.8300572