Neural Machine Translation with Phonetic Embedding

Fangying Zhan
fzhan2@jhu.edu

1. Introduction

From a linguistic point of view, many linguists have agreed that natural language is a natural phenomenon where the study is about the sequences of sounds. However, from an engineering point of view, we are tackling machine translation problems starting by defining languages as sequences of strings and characters. The critical difference lies in the representations of verbal communication. We can either use characters from the writing systems of any language or use other pronunciation units or transliteration to represent languages. It also seems that we assumed the sequences of characters from the writing systems of all languages are already good representations of the languages themselves. To fill in the gap between languages as sounds and languages as strings, in this project, I would like to propose a method to represent languages using both the original texts and the international phonetic alphabet (IPA) as the input data for neural machine translation (NMT) models. Specifically, by focusing on spoken language corpus (text transcriptions), we might recover the phonetic representations from their textual representations by using specific tools and letting models learn from the phonetic information. There is no guarantee that the phonetic approach will be helpful. Still, it is intriguing for us to compare the performance with standard methods and analyze the use of phonetic information in specific tasks.

2. Related Works

A study conducted by Liu et al. shows that the phonetic information combined with textual information helps improve the robustness of NMT in their experiments.[1] Therefore, in this project, I explore the use of phonetic information under similar settings of NMT.

Many works in the past focused on improving MT tasks using a phonetic approach. In tackling the multilingual multimodal NMT task, Chakravarthi et al. observed that using phonetic transcription improves the translation quality after experimenting with Dravidian languages [2]. A phonetic approach is also used to solve out-of-vocabulary words (OOVs) problems, specifically with User Generated Content (UGC) [3], borrowed words (loanwords) [4], and low-resource languages [5].

The grapheme-to-phoneme (g2p) conversion is a critical step involved in a phonetic approach. Deri and Knight used Wiktionary to develop g2p models and adapted high-resource language models to create low-resource ones [6]. Peters et al. proposed a multilingual neural approach to g2p, which outperforms an approach to adapt monolingual models to other low-resource languages [7].

3. Proposed Method

To use the phonetic information in the original NMT setting with textual data only, we need to make use of the embedding of the phonetic representations. Liu et al. proposed the joint embedding method for their robust NMT task, which corresponds to the idea in this project.

3.1. Joint Embedding

The idea is simple and essential: to combine the embedding of the word and its pronunciation sequences. Specifically, for a word a, we obtain the word embedding $\pi(a)$ as usual. In addition, we can also obtain the phonetic embedding $\pi(\psi(a))$, where $\psi(a)$ is the pronunciation sequence of the word a. To achieve this, we need to retrieve the pronunciation in IPA from the input textual data. This can be done via g2p conversions using specific tools available in Python.

The joint embedding for word a is computed as follows:

$$\pi([a, \psi(a)]) = (1 - \beta) \cdot \pi(a) + \beta \cdot \pi(\psi(a)) \qquad (1)$$

where β is a new hyperparameter that we can experiment with to observe the best balance of the original textual information and the added phonetic information that we needed for our tasks.

4. Experiments

4.1. Models

The basic NMT model with attention is used for this project, which has been implemented in our Homework 4 and 5. In the experiments, PyTorch-LSTM is used, with parameters set as follows: hidden size 512, learning rate 2e-4, epochs for training 100,000, and dropout rate 0.1.

4.1.1 LSTM Encoder-Decoder with Attention

To apply the joint embedding method, we are able to modify both the embedding in the encoder and the embedding in the decoder. For this project, the joint embedding is only used in the encoder part, which is expected to give reasonable results if the phonetic approach works well.

An LSTM with a forget gate is implemented using the following equations:

$$f_t = \sigma_g(W_f x_t + U_f h_{t-1} + b_f) \tag{2}$$
$$i_t = \sigma_g(W_i x_t + U_i h_{t-1} + b_i) \tag{3}$$
$$o_t = \sigma_g(W_o x_t + U_o H_{t-1} + b_o) \tag{4}$$
$$\tilde{c}_t = \sigma_c(W_c x_t + U_c h_{t-1} + b_c) \tag{5}$$
$$c_t = f_t \circ c_{t-1} + i_t \circ \tilde{c}_t \tag{6}$$
$$h_t = o_t \circ \sigma_c(c_t) \tag{7}$$

where the parameters are the matrices W_f, W_i, W_o, W_c, U_f, U_i, U_o and U_c, which contain the weights of forget, input, output and memory cell state connections respectively, and the vector b, which contains the bias. The subscripts are the input gate i, the output gate o, the forget gate f, and the memory cell c. Specifically, at each time step t, the variable x_t denotes the input vector to the LSTM unit, f_t denotes the forget gate's activation vector, i_t denotes the input/update gate's activation vector, o_t denotes the output gate's activation vector, h_t denotes the hidden state vector (also known as output vector of the LSTM unit), $\tilde{c}_t$ denotes the cell input activation vector, and c_t denotes the cell state vector. The activation functions used are σ_g, the sigmoid function, and σ_c the hyperbolic tangent function.

To implement the attention mechanism, we need to compute the associations between the last hidden state of the decoder and the encoder states (word representations), normalize the attention values and weigh the contribution of the input word representation h_j to the context vector c_i as follows:

$$a(s_{i-1}, h_j) = v_a^T \tanh(W_a s_{i-1} + U_a h_j) \tag{8}$$
$$\alpha_{ij} = \frac{\exp(a(s_{i-1}, h_j))}{\sum_k \exp(a(s_{i-1}, h_k))} \tag{9}$$
$$c_i = \sum_j \alpha_{ij} h_j \tag{10}$$

where the s_{i-1} denotes the previous hidden state of the decoder, W_a, U_a, v_a denotes the weight matrices and vector respectively, j denotes the input word, i denotes the produced output word, h_j denotes the input word representation, and c_i denotes the context vector.

4.2. Translation Task

The joint embedding method is evaluated on both Chinese-English and French-English translation tasks. For Chinese-English translation, the TVsub (DCU-Tencent Chinese-English Dialogue Corpus) is used for the experiments, which contains 1.23M sentence pairs after filtering and preprocessing for the experiments. A subtitle corpus is chosen because this project is motivated by spoken languages rather than written languages. Yet, subtitle corpora tend to be unclean to some extent, which results in a defect in the experiments. Since Liu et al. also experimented with the Chinese-English translation task, we may still expect that the phonetic approach works for this project as well[1].

The joint embedding method is also evaluated on the French-English Byte Pair Encoding (BPE) dataset from Homework 4 and 5. If the phonetic approach generalizes well to all translation tasks, we expect it also gives better performance on our familiar task.

For training, tuning, and testing purposes, the dataset is shuffled and randomly split into training, development, and test sets. For the TVsub corpus, a reasonable ratio for splitting would be 8:1:1. However, for the experiments, the sizes of dev and test sets are both 400. This could be improved if we obtain a larger and cleaner corpus and apply methods like BPE to the raw corpus to improve the performance.

4.3. Software

The key Python libraries involved are PyTorch and NLTK. Another necessary step is transcribing languages from the input data into IPA, also known as g2p. For this project, the g2p tool Epitran is used, which is available for 69 languages, including Chinese and French[8].

4.4. Evaluation Metrics

The evaluation metric for the translation quality is the 4-gram BLEU score, which is computed as follows:

$$\text{BLEU} = \min(1, \frac{\text{output} - \text{length}}{\text{reference} - \text{length}})(\prod_{i=1}^{4} \text{precision}_i)^{\frac{1}{4}} \tag{11}$$

4.5. Quantitative Results

Table 1. Results on the dev & test sets.

Model (β)	zh-en dev	zh-en test	fr-en dev	fr-en test
LSTM $\beta = 0$	0.08		0.54	
$\beta = 0.3$	**0.09**	0.09	0.57	
$\beta = 0.5$	0.08		0.57	
$\beta = 0.7$	0.08		**0.58**	NA
$\beta = 0.9$	0.07		0.57	

The results of the experiments on both Chinese-English and French-English tasks are shown in Table1. When $\beta = 0$, only text embedding is used, so it is equivalent to the original NMT setting, which is our baseline model. Compared with the baseline, when $\beta = 0.3, 0.5, 0.7, 0.9$, the BLEU scores tend to improve. For the Chinese-English task, the

BLEU scores are poor overall, but the best score is observed when setting $\beta = 0.3$. For the French-English task with BPE sentences, the best BLEU score is observed at $\beta = 0.7$, and the improvement from the baseline is significant.

5. Discussion

The joint embedding is considered as a simple and easy approach to implement. From this project, we can conclude that it significantly improves the translation quality on clean and BPE datasets, including the BPE French-English corpus. However, problems remain unsolved when applying the method to unclean, non-BPE datasets, including the TV-sub corpus. The translation quality remains poor with the basic NMT model. Yet, we can still observe slight improvement when considering the phonetic information. Overall, the joint embedding is expected to work for many translation tasks based on our experiments. To further improve the translation quality, models including the transformer need to be taken into consideration, and more subtle tokenization methods need to be used. Other ways of defining the phonetic embedding may also be further experimented.

References

[1] Hairong Liu, Mingbo Ma, Liang Huang, Hao Xiong, and Zhongjun He. Robust neural machine translation with joint textual and phonetic embedding. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3044–3049. Association for Computational Linguistics.

[2] Bharathi Raja Chakravarthi, Ruba Priyadharshini, Bernardo Stearns, Arun Jayapal, Sridevy S, Mihael Arcan, Manel Zarrouk, and John P McCrae. Multilingual multimodal machine translation for dravidian languages utilizing phonetic transcription. In *Proceedings of the 2nd Workshop on Technologies for MT of Low Resource Languages*, pages 56–63. European Association for Machine Translation.

[3] José Carlos Rosales Núñez, Djamé Seddah, and Guillaume Wisniewski. Phonetic normalization for machine translation of user generated content. In *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*, pages 407–416. Association for Computational Linguistics.

[4] Yulia Tsvetkov and Chris Dyer. Lexicon stratification for translating out-of-vocabulary words. 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)*, pages 125–131. Association for Computational Linguistics.

[5] Khan Md. Anwarus Salam, Setsuo Yamada, and Tetsuro Nishino. Sublexical translations for low-resource language. In *Proceedings of the Workshop on Machine Translation and Parsing in Indian Languages*, pages 39–52. The COLING 2012 Organizing Committee.

[6] Aliya Deri and Kevin Knight. Grapheme-to-phoneme models for (almost) any language. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 399–408. Association for Computational Linguistics.

[7] Ben Peters, Jon Dehdari, and Josef van Genabith. Massively multilingual neural grapheme-to-phoneme conversion. In *Proceedings of the First Workshop on Building Linguistically Generalizable NLP Systems*, pages 19–26. Association for Computational Linguistics.

[8] David R. Mortensen, Siddharth Dalmia, and Patrick Littell. Epitran: Precision G2P for many languages. In Nicoletta Calzolari (Conference chair), Khalid Choukri, Christopher Cieri, Thierry Declerck, Sara Goggi, Koiti Hasida, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, Hélène Mazo, Asuncion Moreno, Jan Odijk, Stelios Piperidis, and Takenobu Tokunaga, editors, *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Paris, France, May 2018. European Language Resources Association (ELRA).