Improving the neural network-based machine transliteration for low-resourced language pair

Ngoc Tan Le
Universite du Quebec a Montreal /
201, avenue du President-Kennedy,
Montreal, Canada
le.ngoc_tan@uqam.ca

Fatiha Sadat
Universite du Quebec a Montreal /
201, avenue du President-Kennedy,
Montreal, Canada
sadat.fatiha@uqam.ca

Abstract

Grapheme-to-phoneme models are key components in automatic speech recognition and text-to-speech systems. With low-resourced language pairs that do not have available and well-developed pronunciation lexicons, grapheme-to-phoneme models are particularly useful. The current work presents an approach that applies an alignment representation for input sequences and pre-trained source and target embeddings to overcome the transliteration challenge for a low-resourced languages pair. The proposed method is tested with French and Vietnamese low-resourced language pair. The results showed promising improvement compared to the state-of-the-art approaches, with a large increase of $+7.30$ BLEU and a reduction in translation error rate (TER) of -8.16 and phoneme error rate (PER) of -14.17.

1 Introduction

In many domains, including machine translation, end-to-end deep learning models have become a valuable alternative to more traditional statistical approaches (Wu et al., 2016; Koehn, 2017). This is our motivation for applying a similar approach to build a machine transliteration system.

In the state-of-the-art, the grapheme-to-phoneme methods were based on the use of grapheme-phoneme mappings (Oh et al., 2006; Duan et al., 2016). However, recurrent neural networks approaches do not require any alignment information. In this study, we propose a method to build a low-resourced machine transliteration system, using RNN-based models and alignment information for input sequences. We are interested in solving out-of-vocabulary words for machine translation systems, such as proper nouns or technical terms, for a low-resourced language pair.

The structure of the article is as follows: Section 2 presents the state of the art on machine transliteration. In section 3, we describe some background and our proposed approach. Then, in section 4, we present several experiments, the evaluations and an error analysis. Finally, section 5 concludes with some perspectives.

2 Related work

Transliteration can be considered as a subtask of machine translation, when we need to translate source graphemes into target phonemes. In other words, an alignment model needs to be constructed first, and the translation model is built on the basis of the alignments. Transliterating a word from the language of its origin to a foreign language is called *Forward Transliteration*, while transliterating a loan-word written in a foreign language back to the language of its origin is called *Backward Transliteration* (Karimi et al., 2011).

Statistical techniques based on large parallel transliteration corpora work well for rich-resource languages but low-resource languages do not have the luxury of such resources. For such languages, rule-based transliteration is the only viable option.

From 2009 to 2016, various transliteration systems were proposed during the Named Entities

Workshop evaluation campaigns[1] (Duan et al., 2016). These campaigns consist in transliterating from English into languages with a wide variety of writing systems, including Hindi, Tamil, Russian, Kannada, Chinese, Korean, Thai and Japanese. We can see that the romanization of non-Latin writing systems remains a complex computational task that depends crucially on which language is involved. Through this workshop, much progress has been made in methodologies for resolving the transliteration of proper nouns. We see the emergence of different approaches, such as grapheme-to-phoneme conversion (Finch and Sumita, 2010; Ngo et al., 2015), based on statistics like machine translation (Laurent et al., 2009; Nicolai et al., 2015) and neural networks (Finch et al., 2016; Shao and Nivre, 2016; Thu et al., 2016). Other works used attentionless sequence-to-sequence models for the transliteration task (Yao and Zweig, 2015; Rosca and Breuel, 2016). One study used a bidirectional Long Short-Term Memory (LSTM) models together with input delays for grapheme-to-phoneme conversion (Rao et al., 2015).

Another important challenge with the extraction of named entities and automatic transliteration is related to the vast variety of writing systems. All these difficulties are aggravated by the lack of bilingual pronunciation dictionaries for proper nouns, ambiguous transcriptions and orthographic variation in a given language. In addition to transliteration generation systems, there are also transliteration mining systems that try to obtain parallel transliteration pairs from comparable corpora (Kumaran et al., 2010; Tran et al., 2016; Sajjad et al., 2017).

In our literature review, we found a few cases in which Vietnamese had been studied for the transliteration task. Cao et al. (2010) applied the statistical-based approach as machine translation in the transliteration task for the English-Vietnamese low-resource language pair, with a performance of 63 BLEU points. Ngo et al. (2015) proposed a statistical model for English and Vietnamese, with a phonological constraint on syllables. Their system performed better than the rule-based baseline system, with a 70% reduction in error rates. Le and Sadat (2017) explored RNN, particularly, LSTM, in the transliteration task for French and Vietnamese. Their results showed that the RNN-based system performed better than the baseline system, which was based on a statistical approach. In this work, we propose a new approach by using alignment representation for input sequences and pre-trained source/target embeddings in the input layer in order to build a neural network-based transliteration system to solve the problem of scattered data due to a low-resource language.

3 Methodology

3.1 Phonology of Vietnamese

French syllable structure is very rich, with a variety of structures such as CV, CVC, $CCVCC$, etc., where C is a consonant and V is a vowel. On the other hand, the structure of syllables in Vietnamese is very simple. One of the linguistic features of Vietnamese is that a word consists of one or more syllables (Phe, 1997). A syllable in Vietnamese has the following structure:

$$Syllable = Onset + Vowel + Coda$$

The boundary of a syllable depends on consonant groups (*onset and coda*) and vowels. Vietnamese uses a Latin alphabet with 29 letters (Figure 1). There are 12 vowels and 17 consonant unigrams, 9 consonant bi-grams and 1 tri-gram. The vowels consist of $V = \{$*“a”, “ă”, “â”, “e”, “ê”, “i”, “o”, “ô”, “ơ”, “u”, “ư”, “y”* $\}$. Possible consonants in the $Onset = \{$*“b”, “ch”, “c”, “d”, “đ”, “gi”, “gh”, “g”, “h”, “kh”, “k”, “l”, “m”, “ngh”, “ng”, “nh”, “n”, “ph”, “q”, “r”, “s”, “th”, “tr”, “t”, “v”, “x”, “p”*$\}$. Of these consonants, eight can appear in $Coda = \{$*“c”, “ch”, “m”, “n”, “ng”, “nh”, “p”, “t”*$\}$. Phe (1997) found about 10,000 syllables in Vietnamese . In this work, we focus on the graphemes and phonemes of all the words in the bilingual pronunciation dictionary.

3.2 Proposed Approach

Our proposed approach for an efficient transliteration consists of three main steps: *(1) pre-processing, (2) modification of the input sequences based on alignment representation and (3) creation of an RNN-based machine transliteration*. The whole process is illustrated in Figure 2.

[1] http://workshop.colips.org/news2016/

Latin alphabet

a b c d e f g h i j k l m n o p q r s t u v w x y z

Vietnamese alphabet

a ă â b c d đ e ê g h i k l m n o ô ơ p q r s t u ư v x y

9 consonant bi-grams and 1 tri-gram

ch gh gi kh ng nh ph th tr ngh

Figure 1: Illustration of Vietnamese alphabet.

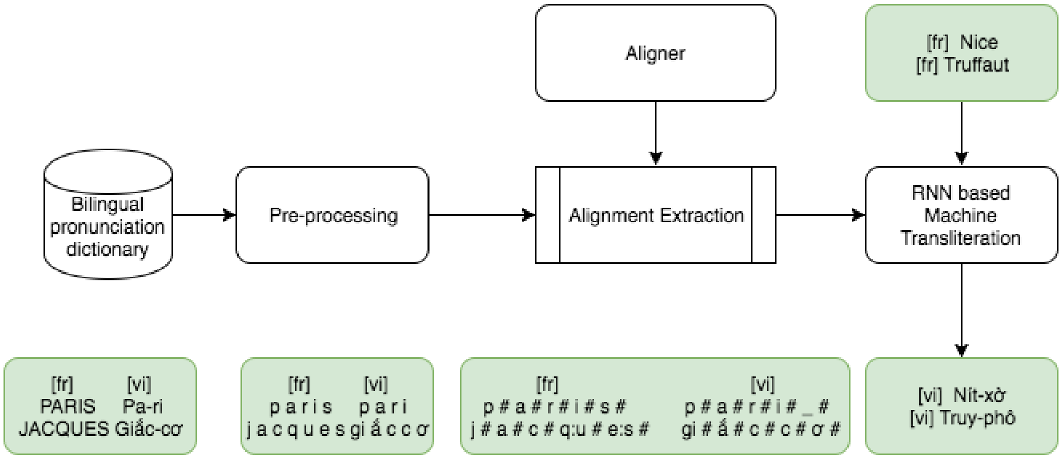

Figure 2: Architecture of machine transliteration for a low-resource language pair dealing with bilingual named entities

(1) First, we obtain a bilingual pronunciation dictionary for a low-resource language pair, in this case French and Vietnamese. Then, this learning data is pre-processed with normalization in lowercasing, removing the hyphens separating syllables and segmenting all syllables at the character level.

(2) We extract the alignment output from the bilingual pronunciation dictionary and modify the input sequences based on the alignment results (Figure 2).

(3) Then we train an RNN-based machine transliteration.

3.3 Grapheme-to-Phoneme Alignments

Given the grapheme sequence of a word and a phoneme sequence corresponding to its pronunciation, an alignment strategy consists of associating each grapheme with its corresponding phoneme, often of length 1. In other words, the grapheme and phoneme sequences are aligned to form joint grapheme-phoneme units, which are called graphone. Yao and Zweig (2015) have reported that, in these alignments, a grapheme may correspond to a null phoneme, a single phoneme or a compound phoneme, with two phonemes. Some illustrations are given in Table 1.

Given a grapheme sequence $G = \{g_1, g_2, ..., g_N\}$, a corresponding phoneme sequence $P = \{p_1, p_2, ..., p_N\}$, and an alignment A, the posterior probability $p(P|G, A)$ is estimated as follows:

<table>
<tr><th>Alignments</th><th>1-to-1</th><th>1-to-2</th><th>2-to-1</th><th>m-to-m</th></tr>
<tr><td><b><i>Graphemes</i></b></td><td>p | a | r | i | s |</td><td>n | i | c | e |</td><td>j | a | c | q:u | e:s |</td><td>t:r | u | f:f | a:u | t |</td></tr>
<tr><td><b><i>Phonemes</i></b></td><td>p | a | r | i | _</td><td>n | í:t | x | ờ |</td><td>gi | ắ | c | c | ơ |</td><td>t:r | u:y | p:h | ô | _ |</td></tr>
</table>

Table 1: Examples of grapheme-to-phoneme alignments, in this case grapheme for French and phoneme for Vietnamese. A grapheme G aligns with a single phoneme (1-to-1), a compound phoneme (1-to-2), two graphemes with one phoneme (2-to-1), or many graphemes with many phonemes (m-to-m). The letter 's' is aligned with a null phoneme '_' that is not pronounced.

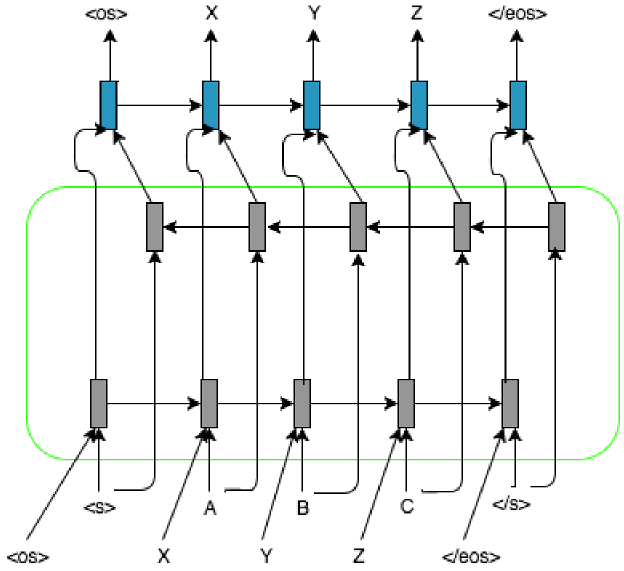

Figure 3: Our RNN-based model architecture with encoder-decoder bi-directional LSTM and alignment representation on input sequences. We use <s> and </s>, <os> and </eos> markers to pad the grapheme/phoneme sequences to a fixed length.

$$p(P|G,A) \approx \prod_{n=1}^{N} p(p_n|p_{n-1}^{n-k}, g_{n-k}^{n+k}) \qquad (1)$$

where k is the context window size, and n is the position index in the alignment.

A graphone alignment strategy can be automatically learnt from a pronunciation dictionary using maximum entropy derived from equation 1 (Barros and Weiss, 2006) or expectation-maximization (Jiampojamarn et al., 2007). Bisani and Ney (2008) used an alignment strategy in a multi-joint sequence model for grapheme-to-phoneme conversion. Their system performed with better accuracy in terms of phoneme error rate.

In contrast, in RNN-based models, that alignment information is not available during the decoding phase, in which the given grapheme sequence does not indicate any clusters aligned with a null/single/compound phoneme or vice-versa. We observe that combining 1-to-2 and 2-to-1 alignment strategies, called bigram-align, can solve this challenge when dealing with small data. Using graphone bigrams can sufficiently cover the output alphabet in the training data. Alignments using graphone bigrams are similar to parallel corpora alignments.

In this work, we use alignment information to alter input sequences because in low-resource settings, representation is crucial for neural networks models. On the other hand, the source and target embeddings will be pre-trained with the training data. For the transliteration task, these embeddings are considered as linear vector mappings between the source and the target. Then they become one of features in the input layer. We expect that exploiting this kind of alignment representation for RNN-based machine transliteration will enhance the system's performance.

4 Experiments

4.1 Configuration

To evaluate the efficiency of our proposed transliteration system in low resource settings, we used a bilingual pronunciation dictionary that has been collected from the news websites, as presented by Cao et al. (2010). The learning data comprise 4,259 pairs of bilingual French-Vietnamese named entities pairs, with a set of vocabularies that contains 31 graphemes on the French source side, and 71 phonemes on the Vietnamese target side. This bilingual dictionary was filtered out from the 146M-word corpus collected from French-Vietnamese newspaper texts available on the Internet between April

2008 and October 2009. We found that most of the named entities were persons, locations and organizations. To overcome the problem of the scattering of learning data, we performed the pre-processing step with segmentation of all syllables at the character level and presented the whole dataset in lowercase. The bilingual pronunciation dictionary for the learning was divided into training, development and test sets at a ratio of 90% , 5% and 5% respectively.

To deal with the alignment representation, we used the *m-2-m aligner*[2] toolkit (Jiampojamarn et al., 2007) to align the training data at the character level. We chose $m = 2$ (bigram-align) for all experiments; this means that a maximum of two graphemes on the source side will be aligned with a maximum of two phonemes on the target side. For the pre-trained source and target embeddings, we applied the *word2vec*[3] toolkit (Mikolov et al., 2013) with a dimension of 64, a continuous space window size of 5 and the *skip-gram* option.

We applied the *nmt-keras*[4] toolkit (Peris, 2017) to train our transliteration model for the French-Vietnamese language pair. In the transliteration system configuration, we used two-layer encoder-decoder bi-directional RNN, with a 64-dimension projection layer to encode the input sequences and 128 nodes in each hidden layer. We performed two mechanisms of LSTM and GRU (*Gated Recurrent Unit*). The attention mechanism focuses on input sequence of annotations (Bahdanau et al., 2014). These 64-dimension character embeddings are shared across encoder and decoder LSTMs layers. We used the *Adam* optimizer to learn the weights of the network with a default learning rate of 0.001. For decoding, the beam search was assigned the size of 6. All the RNN hyper-parameters were determined by tuning on the development set.

4.2 Evaluation

We use different evaluation metrics such as *BiLingual Evaluation Understudy (BLEU)* (Papineni et al., 2002), *Translation Error Rate (TER)* (Snover et al., 2009), and *Phoneme Error Rate (PER)*, that is similar to *Word Error Rate*. These metrics were automatically evaluated with a tool, *MultEval* version 0.5.1[5] (Clark et al., 2011).

To evaluate our proposed approach, we implemented five systems (Table 2):

(1) Baseline system A : *phrase-based statistical machine translation (pbSMT).*
We implemented a *pbSMT* system with *Moses*[6] (Koehn et al., 2007). We used *mGIZA* (Gao and Vogel, 2008) to align the corpus at the character level, and *SRILM* (Stolcke and others, 2002) to create a character-based 5-gram language model for the target language.

(2) Baseline system B : *multi-joint sequence model for grapheme-to-phoneme convertion.*
We applied the *Sequitur-G2P*[7] toolkit to train a transliteration model.

(3) System 1 : *encoder-decoder bidirectional LSTM + attention mechanism.*

(4) System 2 : *encoder-decoder bidirectional GRU + attention mechanism + alignment representation for input sequences.*

(5) System 3 : *encoder-decoder bidirectional GRU + attention mechanism + alignment representation for input sequences + pre-trained source and target embeddings.*

Experiments	**BLEU ↑**	**TER ↓**	**PER ↓**
Baseline A	61.30	24.08	44.20
Baseline B	65.70	20.50	38.00
System 1	66.68	16.70	31.50
System 2	67.57	16.23	30.63
System 3	**68.60**	**15.92**	**30.03**

Table 2: Evaluation of scoring for all systems: BLEU, TER and PER.

The difference between the two baseline systems' performance is minor. Baseline system B seems slightly more efficient than baseline system A, with

[2] https://github.com/letter-to-phoneme/m2m-aligner/

[3] https://code.google.com/archive/p/word2vec/

[4] https://github.com/lvapeab/nmt-keras/

[5] http://www.cs.cmu.edu/~jhclark/downloads/multeval-0.5.1.tgz

[6] http://www.statmt.org/moses/

[7] https://www-i6.informatik.rwth-aachen.de/web/Software/g2p.html

a gain of +4.40 BLEU, as well as reduced -3.58 for translation error rate (TER) and -6.20 for phoneme error rate (PER) (Table 2).

In addition, by comparing the two basic systems and the three systems 1, 2 and 3 (*proposed approach*), we find significant results with scores up to 68.60 BLEU, reductions in translation error rate (TER) and phoneme error rate (PER) up to 15.92 and 30.03, respectively .

In principle, the GRU cell does not need to use a memory unit to control the flow of information like the LSTM cell. It is possible that all hidden states are directly used without any control. The GRU-based architectures have fewer parameters and can therefore train a little faster or need less data to generalize. This is one of the reasons that the system 3 has achieved the best performance.

Otherwise, by comparing the system 3 with the other two systems A and B, we observe significant gains of +7.30 and + 2.90 BLEU as well as reductions of -8.16 and -4.58 TER, -14.17 and -7.97 PER, respectively (Table 2).

All the experimental results showed that using the alignment representation, combining with the pre-trained source and target embeddings resulted in significant advances over other methods.

4.3 Error Analysis

We performed an error analysis of the five evaluation systems to better understand what kinds of errors in predicted phonemes occurred between the French source and the Vietnamese target.

We compared the transliteration prediction results for the named entities in the five evaluation systems with some proper nouns that had not been seen during the learning phase (Table 3).

The results showed that baseline systems A (*pb-SMT*) and B (*multi-joint sequence model*) made some transliteration prediction errors in proper nouns such as Paris, Tours, Truffaut and Zurich, while systems 1, 2 and 3 (*our proposed approach*) provided better results. The baseline systems encountered difficulties in optimally predicting all the transliteration possibilities due to the original variety of proper nouns (*e.g. French, English, Italian, Spanish, Portuguese, Russian, etc.*) and the pronunciation of different tail syllables (Table 3):

- "-s" or "-ce" (xơ or ϕ) : Nice → nít-xơ or Paris → pa-ri
- "-x" or "-ch" (ích or ít or ϕ) : Zurich → giuy-rích or giuy-rít
- and other graphemes such as "-u-" (/y/ = uy or u) : Truffaut → tru-phô or truy-phô

In another analysis, we observed that the baseline systems, especially the statistical-based system A, made some errors such as 'x' to 's' (for Vincente) or 'd uy' to 'iu' (for Yukon) or 'im' to 'anh' (for Zimbabwe), whereas our proposed RNN-based system (system 3) successfully transliterated these cases. The baseline systems removed the last phoneme in the hypotheses for Nice and Jacques, whereas our proposed RNN-based system produced these predictions successfully. But it proposed another transliteration hypothesis in 'b ơ' instead of 'b u ê' (for Zimbabwe) (Table 4).

5 Conclusion

In this paper, we presented an approach for machine transliteration in low resource settings, based on the alignment representation for input sequences and pre-trained source and target embeddings. The method can be trained using a small amount of training data. Eventually, this approach could be extended to any low-resources language pair when a bilingual pronunciation dictionary is available. Therefore, this method is extremely useful for under-resourced languages for which training data is difficult to find.

References

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

Maria João Barros and Christian Weiss. 2006. Maximum entropy motivated grapheme-to-phoneme, stress and syllable boundary prediction for portuguese text-to-speech. *IV Jornadas en Tecnologías del Habla. Zaragoza, Spain*, pages 177–182.

Maximilian Bisani and Hermann Ney. 2008. Joint-sequence models for grapheme-to-phoneme conversion. *Speech communication*, 50(5):434–451.

References	Baseline system A	Baseline system B	System 1	System 2	System 3
p a r i	p a r í t	p a r í t	p a r i	p a r í:t x:ơ	p a r i _
t i g ờ r a n nơ	t i g ờ r a n nơ	t i g ờ r a n nơ	t i g ờ r a n nơ	t i g:ờ r a n n:ơ	t i g:ờ r a n n:ơ
t u l u gi ơ	t u l u x ơ	t u l u x ơ	t u l u gi ơ	t u l uy s:ờ	t u l uy s:ơ
t u a	t u ố c t x ơ	t u ố c t x ơ	t uý t xơ	t:ơ uy _ x:ơ	t u a _ x:ơ
tr uy ph ô	tr uy ph ố t	tr uy ph ố t	tr ú p ph ô	tr uy ph ô	tr u ph ô
gi uy r í ch	gi uy r i	gi uy r i	gi uy r í ch	gi uy r í:ch	gi uy r í:t

Table 3: Examples of transliteration prediction results by all systems, with IPA (*International Phonetic Alphabet*) transcriptions, hypothesis and reference (ground truth) for six proper nouns: *Paris, Tigrane, Toulouse, Tours, Truffaut and Zurich*.

System	Hypothesis	Reference
Baseline A	p a r í t	p a r i
	n í t	n í t x ờ
	gi ắ c	gi ắ c c ơ
	v a nh s â n t ơ	v a nh x â n t ơ
	iu k ô ng	d uy c ô ng
	gi anh b a b u ê	gi i m b a b u ê
System 3	p a r i	p a r i
	n í t x ờ	n í t x ờ
	gi ắ c c ơ	gi ắ c c ơ
	v a nh x â n t ơ	v a nh x â n t ơ
	d uy c ô ng	d uy c ô ng
	gi i m b a b ơ	gi i m b a b u ê

Table 4: Examples of transliteration prediction results by two systems, with hypothesis and reference (as ground truth) for six proper nouns: *Paris, Nice, Jacques, Vincente, Yukon and Zimbabwe*.

Nam X Cao, Nhut M Pham, and Quan H Vu. 2010. Comparative analysis of transliteration techniques based on statistical machine translation and joint-sequence model. In *Proceedings of the 2010 Symposium on Information and Communication Technology*, pages 59–63. Association for Computing Machinery.

Jonathan H Clark, Chris Dyer, Alon Lavie, and Noah A Smith. 2011. Better hypothesis testing for statistical machine translation: Controlling for optimizer instability. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*, pages 176–181. Association for Computational Linguistics.

Xiangyu Duan, Rafael E Banchs, Min Zhang, Haizhou Li, and A Kumaran. 2016. Report of news 2016 machine transliteration shared task. *ACL 2016*, pages 58–72.

Andrew Finch and Eiichiro Sumita. 2010. Transliteration using a phrase-based statistical machine translation system to re-score the output of a joint multigram model. In *Proceedings of the 2010 Named Entities Workshop*, pages 48–52. Association for Computational Linguistics.

Andrew Finch, Lemao Liu, Xiaolin Wang, and Eiichiro Sumita. 2016. Target-bidirectional neural models for machine transliteration. *ACL 2016*, pages 78–82.

Qin Gao and Stephan Vogel. 2008. Parallel implementations of word alignment tool. In *Software Engineering, Testing, and Quality Assurance for Natural Language Processing*, pages 49–57. Association for Computational Linguistics.

Sittichai Jiampojamarn, Grzegorz Kondrak, and Tarek Sherif. 2007. Applying many-to-many alignments and hidden markov models to letter-to-phoneme conversion. In *HLT-NAACL*, volume 7, pages 372–379.

Sarvnaz Karimi, Falk Scholer, and Andrew Turpin. 2011. Machine transliteration survey. *ACM Computing Surveys (CSUR)*, 43(3):17.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th annual meeting of the ACL on interactive poster and demonstration sessions*, pages 177–180. Association for Computational Linguistics.

Philipp Koehn. 2017. Neural machine translation. *arXiv preprint arXiv:1709.07809*.

A Kumaran, Mitesh M Khapra, and Haizhou Li. 2010. Report of news 2010 transliteration mining shared task. In *Proceedings of the 2010 Named Entities Workshop*, pages 21–28. Association for Computational Linguistics.

Antoine Laurent, Paul Deléglise, Sylvain Meignier, and France Spécinov-Trélazé. 2009. Grapheme to phoneme conversion using an smt system. In *Proceedings of INTERSPEECH, ISCA*, pages 708–711.

Ngoc Tan Le and Fatiha Sadat. 2017. A neural network transliteration model in low resource settings. In

Proceedings of the 16th International Conference of Machine Translation Summit, September 18-22 2017, Nagoya, Japan, volume 1, Research Track, pages 337–345.

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013. Linguistic regularities in continuous space word representations. In *hlt-Naacl*, volume 13, pages 746–751.

Hoang Gia Ngo, Nancy F Chen, Binh Minh Nguyen, Bin Ma, and Haizhou Li. 2015. Phonology-augmented statistical transliteration for low-resource languages. In *Interspeech*, pages 3670–3674.

Garrett Nicolai, Bradley Hauer, Mohammad Salameh, Adam St Arnaud, Ying Xu, Lei Yao, and Grzegorz Kondrak. 2015. Multiple system combination for transliteration. In *Proceedings of NEWS 2015 The Fifth Named Entities Workshop*, pages 72–79.

Jong-Hoon Oh, Key-Sun Choi, and Hitoshi Isahara. 2006. A machine transliteration model based on correspondence between graphemes and phonemes. *ACM Transactions on Asian Language Information Processing (TALIP)*, 5(3):185–208.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

Álvaro Peris. 2017. NMT-Keras. `https://github.com/lvapeab/nmt-keras`. GitHub repository.

Hoang Phe. 1997. Vietnamese dictionary. *Vietnam Lexicography Centre, Da Nang Publishing House*.

Kanishka Rao, Fuchun Peng, Haşim Sak, and Françoise Beaufays. 2015. Grapheme-to-phoneme conversion using long short-term memory recurrent neural networks. In *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*, pages 4225–4229. IEEE.

Mihaela Rosca and Thomas Breuel. 2016. Sequence-to-sequence neural network models for transliteration. *arXiv preprint arXiv:1610.09565*.

Hassan Sajjad, Helmut Schmid, Alexander Fraser, and Hinrich Schütze. 2017. Statistical models for unsupervised, semi-supervised and supervised transliteration mining. *Computational Linguistics*.

Yan Shao and Joakim Nivre. 2016. Applying neural networks to english-chinese named entitytransliteration. In *Sixth Named Entity Workshop, joint with 54th ACL*.

Matthew G Snover, Nitin Madnani, Bonnie Dorr, and Richard Schwartz. 2009. Ter-plus: paraphrase, semantic, and alignment enhancements to translation edit rate. *Machine Translation*, 23(2-3):117–127.

Andreas Stolcke et al. 2002. Srilm-an extensible language modeling toolkit. In *Interspeech*, volume 2002, page 2002.

Ye Kyaw Thu, Win Pa Pa, Yoshinori Sagisaka, and Naoto Iwahashi. 2016. Comparison of grapheme–to–phoneme conversion methods on a myanmar pronunciation dictionary. *Proceedings of the 6th Workshop on South and Southeast Asian Natural Language Processing 2016*, pages 11–22.

Phuoc Tran, Dien Dinh, and Hien T Nguyen. 2016. A character level based and word level based approach for chinese-vietnamese machine translation. *Computational intelligence and neuroscience*, 2016.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Kaisheng Yao and Geoffrey Zweig. 2015. Sequence-to-sequence neural net models for grapheme-to-phoneme conversion. *arXiv preprint arXiv:1506.00196*.