

---

# A Neural Network Transliteration Model in Low Resource Settings

**Tan Le**  
Universite du Quebec a Montreal, Canada

le.ngoc\_tan@courrier.uqam.ca

**Fatiha Sadat**  
Universite du Quebec a Montreal, Canada

sadat.fatiha@uqam.ca

---

## Abstract

Transliteration is the process of converting a text in one script to another, guided by phonetic clues. This conversion requires an important set of rules defined by expert linguists to determine how the phonemes are aligned and to take into account the phonology system of the target language. The problem with under-resourced language pairs remains the lack of linguistic resources. In this research, we present a recurrent neural network based approach to overcome the transliteration problem for a low-resource language pair, with an application on the French-Vietnamese language pair. Our system requires a small bilingual learning dataset. We obtained promising results with a large gain of BLEU-score and a reduction in translation errors rate (TER) and phonemes errors rate (PER), compared to other systems.

## 1 Introduction

Transliteration consists of a process of transforming a word from a writing system (called *source word*) to a phonetically equivalent word of another writing system (called *target word*) (Knight and Graehl, 1998). Many of the named entities (*i.e. person names, location, organization, technical terms, etc.*) are often transliterated from a source language to a target language when translation is difficult or impossible. Transliteration can be considered as a sub-task of machine translation (MT).

Named entities constitute an open morphological class. Person names and organizations names, which are never seen before in the learning phase, often appear in the new documents. It is critical that MT systems address this issue. Integrating a transliteration module within a MT system remains a solution for solving out-of-vocabulary words (OOV) having the type of named entities.

Moreover, with the evolution of high technologies and the globalization of commerce, people tend to invent new words. It is very difficult to define all the possible rules of phonetic transformation between the source language and the target language.

In this research, we propose a method of low resource machine transliteration using recurrent neural network (RNN) based model. This task automatically predicts the phonemic representation of a word in the target language given a new word in the source language that does not exist in the dictionary of bilingual phonetics. We are interested in solving out-of-vocabulary words considered as proper names or technical terms from a machine translation system for a under-resourced language pair, with application for French-Vietnamese.

Our contribution is to show how, with a small bilingual learning dataset, we can train a RNN-based model for low resource machine transliteration. To the problem of sparse data due

to the low resource languages, we apply an algorithm to re-rank the list of  $k$ -best results from the baseline transliteration model.

The structure of the article is as follows: Section 2 presents the state of the art on transliteration. Section 3 describes our proposed approach. Section 4, we present our experiments and compare the performance of our system with other systems as well as errors analysis. Finally, in section 5, we conclude with some perspectives.

## 2 Related work

Since 2009, various transliteration systems have been proposed during the Named Entities Workshop evaluation campaigns <sup>1</sup> (Duan et al., 2016). These campaigns consist of 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 is highly dependent on a language.

Through this workshop, much progress has been made in the methodologies with an emergence of different approaches, such as grapheme in the phoneme (Finch and Sumita, 2010; Ngo et al., 2015), based on statistics like automatic translation (Laurent et al., 2009; Nicolai et al., 2015) as well as neural networks (Finch et al., 2015, 2016; Shao and Nivre, 2016; Thu et al., 2016).

The variety of writing systems adds another important challenge in the extraction of named entities and automatic transliteration. All these difficulties are aggravated by the lack of bilingual dictionaries of proper names, ambiguities of transcription as well as orthographic variation in a language.

(Lo et al., 2016) used a semi-supervised transliteration model built on a seed corpus mined from the standard parallel training data, in order to improve the Russian-English machine translation system for WMT 2016.

(Ngo et al., 2015) proposed a statistical model for a language pair with English-Vietnamese language, with a phonological constraint on the syllables. Their system has achieved better performance than the base system, based on rules, with a 70% reduction in error rates.

(Cao et al., 2010) also applied the statistical-based approach as automatic translation in the transliteration task for a language pair with little English-Vietnamese language, with a performance of 63% of BLEU (Papineni et al., 2002). Our proposed approach is totally different, except for the same preparation of the bilingual phonetic dictionary learning. We propose a step of rescoring  $k$ -best results from the baseline transliteration system to solve the problem of scattered data due to the low resource language.

## 3 Proposed Approach

### 3.1 Phonology of Vietnamese

The structure of syllables in French 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 peculiarities of Vietnamese is that a word consists of a syllable or several syllables (Phe, 1997). A syllable in Vietnamese is constituted with the following structure:

$$\text{Syllable} = \text{Onset} + \text{Vowel} + \text{Coda}$$

The boundary of a syllable depends on consonant groups (*onset and coda*) and vowels. The Vietnamese has a Latin alphabet with 29 letters. There are 12 vowels and 17 consonants uni-grams, 9 consonants bi-grams and 1 tri-gram. The vowels are  $V = \{“a”, “\grave{a}”, “\hat{a}”, “e”, “\hat{e}”,$

<sup>1</sup><http://workshop.colips.org/news2016/>

“l”, “o”, “o”, “o”, “u”, “u”, “y”}. The consonants are  $Onset = \{“b”, “ch”, “c”, “d”, “d”, “gi”, “gh”, “g”, “h”, “kh”, “k”, “l”, “m”, “ngh”, “ng”, “nh”, “n”, “ph”, “q”, “r”, “s”, “th”, “tr”, “t”, “v”, “x”, “p”\}$ . Among these consonants, there are 8 in tail  $Coda = \{“c”, “ch”, “n”, “nh”, “ng”, “m”, “p”, “t”\}$ . The Vietnamese has 6 lexical tones such as *up* (i.e. *có* = *own*), *broken* (i.e. *mỹ* = *american*), *flat* (i.e. *ba* = *father*), *interrogative* (i.e. *thuỷ* = *water*), *down* (i.e. *trà* = *tea*) and *low* (i.e. *lại* = *coming*). (Phe, 1997) found about 10,000 syllables for Vietnamese. In this research work, we focus only on the grapheme and the phoneme of all words in the bilingual dictionary.

### 3.2 Multi-joint sequence model

The approach of graphemes-to-phonemes with a multi-joint sequence model has been proposed by (Deligne et al., 1995). This is one of the most popular approaches in the task of converting graphemes into phonemes by machine learning. The main idea consists in generating both the sequences at the level of graphemes and at the level of phonemes by means of a single joined sequence of the linguistic units which represent all the symbols of graphemes and phonemes. In fact, the aim of this approach is to find a sequence of phonemes  $Y$  defined by  $Y = P_1^m = \{p_1, p_2, \dots, p_m\}$ , Given a sequence of graphemes  $X$  defined by  $X = G_1^m = \{g_1, g_2, \dots, g_m\}$ . The problem can become the estimation of the most optimal  $Y$  phoneme sequence, which maximizes their conditional probability as in the following equation 1:

$$\hat{e} = \underset{Y}{\operatorname{argmax}} p(Y|X) \quad (1)$$

According to the Bayes' Theorem:

$$\hat{e} = \underset{Y}{\operatorname{argmax}} \frac{p(X|Y) p(Y)}{p(X)} \quad (2)$$

Because  $p(X)$  is independent of all the phoneme sequences  $Y$ , the equation 2 can be simplified as follows:

$$\hat{e} = \underset{Y}{\operatorname{argmax}} p(X|Y) p(Y) \quad (3)$$

### 3.3 Recurrent neural network based sequence model for small data

Figure 1 shows the architecture of a RNN model adapted from (Yao and Zweig, 2015). A RNN model takes an input of a sequence of vectors  $(x_1, x_2, \dots, x_n)$  and produces an output of a sequence of vectors  $(h_1, h_2, \dots, h_n)$  to represent the information at each input step. LSTMs (*Long-Short Term Memory*), which are a type of RNNs, have been designed to incorporate a memory cell which can protect and control the cell state. They use several gates to control the amount of information from the previous states which should be forgotten and the information from the inputs which should be updated to the memory cell (Hochreiter and Schmidhuber, 1997). The formulas that govern the computation happening in a RNN are as follows:

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i) \quad (4)$$

$$c_t = (1 - i_t) \odot c_{t-1} + i_t \odot \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \quad (5)$$

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o) \quad (6)$$

$$h_t = o_t \odot \tanh(c_t) \quad (7)$$

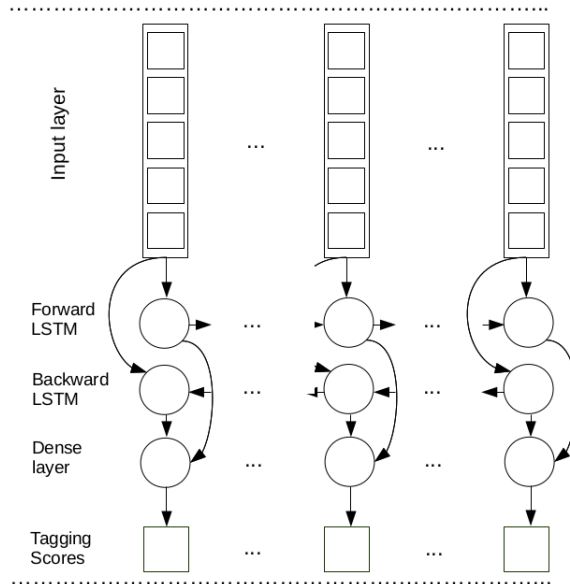


Figure 1: Our recurrent neural network based model architecture

where  $\sigma$  is the element-wise sigmoid function, and  $\odot$  is the element-wise product.  $c_t$  and  $o_t$  are the cell state and the output at the step  $t$ , respectively.

There are many variants of LSTM implementations. LSTM sequence-to-sequence models were successfully applied in various tasks, including machine translation (Sutskever et al., 2014) and grapheme-to-phoneme (Yao and Zweig, 2015).

Our approach consists of three main steps: (1) *pre-processing*, (2) *creating a RNN-based model*, and (3) *re-ranking the  $k$ -best*. The whole process is illustrated in Figure 1.

- (1) First, we collect bilingual phonetic linguistic resources for a low resource language pair, here French-Vietnamese. Then, this learning data is pre-processed with normalization in miniscule as well as a segmentation of syllables in Vietnamese, which is explained in section 3.1 - phonology of Vietnamese.
- (2) Then we train a RNN-based model.
- (3) Finally, we implement an additional module to re-rank the list of  $k$ -best results from the transliteration model of bilingual proper names. Inspired from (Bhargava et al., 2011), we use the algorithm of *Support Vector Machines (SVM)* for this module, with different characteristics such as *phonemic alignment scores*, *orthographic and phonetic similarities and length difference between each pair of graphemes-phonemes*.

## 4 Experimentation

### 4.1 Data preparation

We use a bilingual phonetic dictionary that has been collected from the news websites as presented in (Cao et al., 2010). The data learning has 4,259 pairs of bilingual French-Vietnamese proper names, with a set of vocabularies that contains 31 graphemes in the French source side, and 71 phonemes in the Vietnamese target side. We find that most of the bilingual proper names are person names, location names and organization names. To overcome the problem of

the scattering of learning data, we perform the pre-processing with normalization for the entire data (Figure 2).

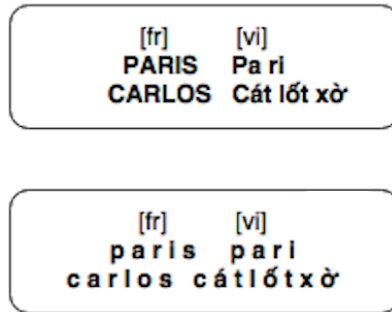


Figure 2: Illustration of the bilingual phonetic dictionary and the pre-processing results

Inspired by the phrases-based statistical approach (*pbSMT*), we consider a baseline system by applying this approach, but based on characters. We implement a *pbSMT* system with the *Moses*<sup>2</sup> (Koehn et al., 2007). We use *mGIZA* (Gao and Vogel, 2008) to align the corpus to the character level, and *SRILM* (Stolcke et al., 2002) to create a 5-gram language model for Vietnamese. While the *pbSMT* systems implemented by (Finch and Sumita, 2010)(Nicolai et al., 2015) have not taken into account word reordering, we will test various word reordering models offered by *Moses*.

We apply *Sequitur-G2P*<sup>3</sup> tool to train our transliteration model of bilingual proper names for the French-Vietnamese language pair.

We used 2-layer bi-directional *Long Short-Term Memory* (LSTM) cells (Hochreiter and Schmidhuber, 1997) for the RNN-based model, with a 64-dimensional projection layer to encode the input sequences and 64 nodes in each hidden layer. We used the '*sgd*' (*Stochastic Gradient Descent*) optimizer to learn the weights of the network with a learning rate of 0.5. We used *g2p-sep2seq*<sup>4</sup> toolkit. This implementation is based on python *TensorFlow*, which allows an efficient training on both CPU and GPU.

We implement a *SVM re-classification module* using the *LinearSVC* library of *scikit-learn*<sup>5</sup> for the purpose of rescoring the best hypotheses from a list of *k*-best (with *k* = 100) results obtained by the baseline transliteration system.

## 4.2 Evaluation

The bilingual phonetic dictionary of learning is split into one training set, one development set and one test set with a ratio of 80 %, 10% and 10 % respectively.

We apply different evaluation metrics such as *BiLingual Evaluation Understudy* (Papineni et al., 2002), *Translation Error Rate (TER)* (Snover et al., 2009) with a tool of *multeval* version 0.5.1<sup>6</sup> (Clark et al., 2011).

For the phonemic error rate, we use a *Phoneme Error Rate (PER)* metric with *SCLITE* (a tool for calculating scores and evaluating the results of speech recognition systems) NIST

<sup>2</sup><http://www.statmt.org/moses/>

<sup>3</sup><https://www-i6.informatik.rwth-aachen.de/web/Software/g2p.html>

<sup>4</sup><https://github.com/cmuspinx/g2p-seq2seq>

<sup>5</sup><http://scikit-learn.org/stable/modules/Generated/sklearn.svm.LinearSVC.html>

<sup>6</sup><http://www.cs.cmu.edu/~jhclark/downloads/multeval-0.5.1.tgz>

SCTK version 2.4.10<sup>7</sup>. The method of calculating the error rate of phonemes with *SCLITE* is similar to that for words (*Word Error Rate*). We use the Levenshtein distance measure in this work. This distance measure is shown in the equation 8, where  $N$  is the number of phonemes, as follows:

$$PER = \frac{\sum_{i=1}^n d_{edit}(hypotheses_i, reference_i)}{|N|} \quad (8)$$

In order to evaluate our proposed approach, we implement three systems, including the baseline system (*pbSMT*), system 1 (*Sequitur-g2p*) and system 2 (*our proposed approach*) (Table 1).

If we compare the baseline system with system 1, the difference in their performance is minor. System 1 seems slightly more efficient than the baseline system, with a gain of +4.40% of BLEU, as well as reductions of -4.30% and -6.20% of translation errors (TER) and phonemes (PER) respectively.

On the other hand, by comparing the baseline system and the system 2 (*our proposed approach*), we note significant results with a gain of +26.95% of BLEU, reductions in translation errors (TER) and phonemes (PER) with -9.30% and -31.30% respectively.

In addition, the performance of system 2 is higher than that of system 1, with a gain of +22.55% of BLEU, reductions of -5.0% and -25.10% of translation error (TER) and phonemic (PER) rates respectively (Table 1).

In general, the proposed approach has achieved very well the transliteration task with the significant gains and can reduce the phoneme error rate. We can observe the output quality of the proposed approach, which is based on the recurrent neural network, is more fluid, coherent and with fewer errors than the baseline and the system 1, which are both based on the statistical approach. We carry out an error analysis on the next section to more details.

Metric	System	Average	$\bar{s}_{sel}$	$s_{Test}$	$p$ -value
BLEU ↑	Baseline ( <i>pbSMT</i> )	61.30	1.70	-	-
	System 1 ( <i>Sequitur-g2p</i> )	65.70	1.70	-	0.79
	System 2 ( <i>our approach</i> )	<b>88.25</b>	1.50	-	0.01
TER ↓	Baseline ( <i>pbSMT</i> )	24.80	1.20	-	-
	System 1 ( <i>Sequitur-g2p</i> )	20.50	1.20	-	0.13
	System 2 ( <i>our approach</i> )	<b>15.50</b>	1.00	-	0.00
PER ↓	Baseline ( <i>pbSMT</i> )	44.20	-	-	-
	System 1 ( <i>Sequitur-g2p</i> )	38.00	-	-	-
	System 2 ( <i>our approach</i> )	<b>12.90</b>	-	-	-

Table 1: Evaluation about scoring for all systems : **BLEU**, **TER** and **PER**.

$p$  -values are relative to the base system and indicate whether a difference of this magnitude between the baseline system and other systems.  $\bar{s}_{sel}$  indicates the variance due to the selection of the test.

### 4.3 Error analysis

We perform an error analysis in the three evaluation systems to better understand the errors likely to predicted phonemes from French to Vietnamese.

First, we check the top-5 best results, for example, from our transliteration model before re-ranking the list of  $k$ -best results (Tables 2 and 3). We find that the first result of transliteration in Vietnamese, having the best probability given a grapheme in French, is not always the correct

<sup>7</sup><http://www1.icsi.berkeley.edu/Speech/docs/sctk-1.2/sclite.htm>

PARIS			MANHATTAN		
No	TOP-5	Probability	No	TOP-5	Probability
1	p a r i	<b>0.633242</b>	1	m a n h á t t â n	0.321082
2	p a r í t	0.153536	2	<b>m a n h á t t a n</b>	<b>0.288677</b>
3	b a r i	0,065151	3	m â n h á t t â n	0.080221
4	b a r í t	0.037314	4	m â n h á t t a n	0.072125
5	p a r í t x ø	0.028526	5	m a h á t t â n	0.058193

Table 2: Illustration of the transliteration predictions of the named entities obtained by our proposed approach before the re-ranking of the list of  $k$ -best results, with the top-5 ( $k = 5$ ) first best results for the named entities : *PARIS* and *MANHATTAN*

GAULOISE		
No	TOP-5	Probability
1	g ô l o a x ø	0.102710
2	g ô n l o a x ø	0.096937
3	<b>g ô l o a d ò</b>	<b>0.092091</b>
4	g ô n l o i d ò	0.086915
5	g ô l o a ò	0.072750

Table 3: Illustration of the transliteration predictions of the named entities obtained by our proposed approach before the re-ranking of the list of  $k$ -best results, with the top-5 ( $k = 5$ ) first best results for the named entities : *GAULOISE*

transliteration. Therefore, it is essential to re-classify the best hypotheses among a list of  $k$ -best (with  $k = 100$ ) results of the transliteration system. For example: *PARIS* -> *p a r i* ( $p = 0.633242$ ), *MANHATTAN* -> *m a n h á t t a n* ( $p = 0.288677$ ) and *GAULOISE* -> *g ô l o a d ò* ( $p = 0.092091$ ).

Evaluation	Proper Names	IPA format	Hypothesis	Reference
<b>Baseline</b>	paris	<b>p a r i</b>	p a r í t	p a r i
	tigrane	<b>t i g r a n e</b>	t i g ò r a n n ø	t i g ò r a n n ø
	toulouse	<b>t u l u s e</b>	t u l u x ø	t u l u g i ø
	tours	<b>t u r</b>	t u ó c t x ø	t u a
	truffaut	<b>t r y f o</b>	t r u y p h ó t	t r u y p h ø
	zurich	<b>z y r i k</b>	g i u y r i	g i u y r í c h
<b>System 1</b>	paris	<b>p a r i</b>	p a r í t	p a r i
	tigrane	<b>t i g r a n e</b>	t i g ò r a n n ø	t i g ò r a n n ø
	toulouse	<b>t u l u s e</b>	t u l u x ø	t u l u g i ø
	tours	<b>t u r</b>	t u ó c t x ø	t u a
	truffaut	<b>t r y f o</b>	t r u y p h ó t	t r u y p h ø
	zurich	<b>z y r i k</b>	g i u y r i	g i u y r í c h
<b>System 2</b>	paris	<b>p a r i</b>	p a r i	p a r i
	tigrane	<b>t i g r a n e</b>	t i g ò r a n n ø	t i g ò r a n n ø
	toulouse	<b>t u l u s e</b>	t u l u x ø	t u l u g i ø
	tours	<b>t u r</b>	t u a	t u a
	truffaut	<b>t r y f o</b>	t r u y p h ø	t r u y p h ø
	zurich	<b>z y r i k</b>	g i u y r í c h	g i u y r í c h

Table 4: Examples of transliteration prediction results by all systems, with IPA (*International Phonetic Alphabet*) format as *ground truth*, hypothesis and reference for six proper names such as *PARIS*, *TIGRANE*, *TOULOUSE*, *TOURS*, *TRUFFAUT* and *ZURICH*

We then perform a comparison of the transliteration prediction results of the named entities between the three evaluation systems with some named entities that are not yet seen during the learning phase (Table 4). We find that the baseline system (*pbSMT*) and system 1 (*Sequitur-G2P*) have incorrectly transliterated the proper names such as PARIS, TOURS, TRUFFAUT and ZURICH, while system 2 (*our proposed approach*) provided good results. We note that the three systems encounter difficulties in predicting optimally all the possibilities of transliteration of bilingual proper names due to the original variety of named entities (*i.e. French, English, Italian, Russian, etc.*) As well as the pronunciation of different tail syllables such as "-er" (/e/ = ê or /e/ = e), "-s" (xσ or φ), "-te" (tσ or φ) or "-x" (ích or φ).

## 5 Conclusion

In this paper, we presented a recurrent neural network based approach to overcome the transliteration problem for a low resource language pair, with an application on the French-Vietnamese language pair. Results show that the RNN-based model outperforms both the phrasal MT and the *Sequitur-G2P* baselines. The RNN-based model yields significant improvements in error rates over state-of-the-art systems.

To our knowledge, we are not aware of any research nor study that analyzes Vietnamese in the transliteration task. Our research focusing on machine transliteration is the first work for the French-Vietnamese bilingual low resource language pair. This system requires only a bilingual phonetic dictionary. This system has the capacity to learn, automatically, the linguistic regularities from this bilingual phonetic dataset.

In future work, we intend to develop our approach with a larger bilingual phonetic dataset as well as to study other approaches such as *attentional mechanism*, in order to improve the performance of neural network models when low amounts of training data are available.

## References

- Bhargava, A., Hauer, B., and Kondrak, G. (2011). Leveraging transliterations from multiple languages. In *Proceedings of the 3rd Named Entities Workshop (NEWS 2011)*, pages 36–40.
- Cao, N. X., Pham, N. M., and Vu, Q. H. (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.
- Clark, J. H., Dyer, C., Lavie, A., and Smith, N. A. (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.
- Deligne, S., Yvon, F., and Bimbot, F. (1995). Variable-length sequence matching for phonetic transcription using joint multigrams. In *Fourth European Conference on Speech Communication and Technology*.
- Duan, X., Banchs, R. E., Zhang, M., Li, H., and Kumaran, A. (2016). Report of news 2016 machine transliteration shared task. *Association for Computational Linguistics 2016*, pages 58–72.
- Finch, A., Liu, L., Wang, X., and Sumita, E. (2015). Neural network transduction models in transliteration generation. In *Proceedings of NEWS 2015 The Fifth Named Entities Workshop*, page 61.
- Finch, A., Liu, L., Wang, X., and Sumita, E. (2016). Target-bidirectional neural models for machine transliteration. *Association for Computational Linguistics 2016*, pages 78–82.



- Finch, A. and Sumita, E. (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.
- Gao, Q. and Vogel, S. (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.
- Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural Comput.*, 9(8):1735–1780.
- Knight, K. and Graehl, J. (1998). Machine transliteration. *Computational Linguistics*, 24(4):599–612.
- Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., 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.
- Laurent, A., Deléglise, P., Meignier, S., and Spécinov-Trélazé, F. (2009). Grapheme to phoneme conversion using an smt system. In *Proceedings of INTERSPEECH, ISCA*, pages 708–711.
- Lo, C.-k., Cherry, C., Foster, G., Stewart, D., Islam, R., Kazantseva, A., and Kuhn, R. (2016). Nrc russian-english machine translation system for wmt 2016. In *Proceedings of the First Conference on Machine Translation, Berlin, Germany. Association for Computational Linguistics*.
- Ngo, H. G., Chen, N. F., Nguyen, B. M., Ma, B., and Li, H. (2015). Phonology-augmented statistical transliteration for low-resource languages. In *Interspeech*, pages 3670–3674.
- Nicolai, G., Hauer, B., Salameh, M., St Arnaud, A., Xu, Y., Yao, L., and Kondrak, G. (2015). Multiple system combination for transliteration. In *Proceedings of NEWS 2015 The Fifth Named Entities Workshop*, pages 72–79.
- Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (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.
- Phe, H. (1997). Vietnamese dictionary. *Vietnam Lexicography Centre, Da Nang Publishing House*.
- Shao, Y. and Nivre, J. (2016). Applying neural networks to english-chinese named entity transliteration. In *Sixth Named Entity Workshop, joint with 54th Association for Computational Linguistics*.
- Snover, M. G., Madnani, N., Dorr, B., and Schwartz, R. (2009). Ter-plus: paraphrase, semantic, and alignment enhancements to translation edit rate. *Machine Translation*, 23(2-3):117–127.
- Stolcke, A. et al. (2002). Srilm-an extensible language modeling toolkit. In *Interspeech*, volume 2002, page 2002.
- Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.
- Thu, Y. K., Pa, W. P., Sagisaka, Y., and Iwahashi, N. (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.
- Yao, K. and Zweig, G. (2015). Sequence-to-sequence neural net models for grapheme-to-phoneme conversion. *arXiv preprint arXiv:1506.00196*.