

A Minimal-Resource Transliteration Framework for Vietnamese

Hoang Gia Ngo¹, Nancy F. Chen², Sunil Sivadas², Bin Ma², Haizhou Li²

¹National University of Singapore, Singapore

²Institute for Infocomm Research, Singapore

ngohoanggia@nus.edu.sg, {nfychen,sivadass, mabin, hli}@i2r.a-star.edu.sg

Abstract

Transliteration converts words in a source language (e.g., English) into phonetically equivalent words in a target language (e.g., Vietnamese). Transliteration is therefore used to handle out-of-vocabulary (OOV) words adopted from foreign languages in automatic speech recognition and keyword search systems. While statistical transliteration approaches have been widely adopted, they may not always be suitable for under-resourced languages, where training data is scarce. In this work, we present a rule-based Vietnamese transliteration framework suitable for spoken language applications with minimal linguistic resources. We show that the proposed system outperforms statistical baselines by up to 81.70% relative when there is limited training examples (94 word pairs). In addition, we investigate the trade-off between training corpus size and transliteration performance of different methods on two distinct corpora. We also show that the proposed model outperforms statistical baselines up to 36.76% relative in keyword search tasks.

Index Terms: subword modeling, grapheme-to-phoneme (g2p), machine translation, low-resourced languages

1. Introduction

In every language, new words are constantly being invented or borrowed from foreign languages (e.g., Facebook has become new vocabulary in non-English languages). These new words present out-of-vocabulary (OOV) challenges to spoken language technologies such as automatic speech recognition (ASR) and keyword search systems. Letter-to-sound tools employing statistical approaches are common solutions for OOV's, though these methods are less effective when training data are limited. OOV's adopted from foreign words are often difficult to model because compared to OOV's originating from the native language, their numbers are smaller in size and the principles governing their pronunciations are intrinsically different. Transliteration is a mechanism for modeling OOV's adopted from foreign languages. Transliteration is the task of converting words written in one writing system (source language, e.g., English) into phonetically equivalent words in another writing system (target language, e.g., Vietnamese) [1], and is often used to translate foreign names of people, location, organizations, products, and concepts. For example, the English word *facebook* can be translated into Vietnamese phonemes: f @I _1 . b_< u k _2 (". " is the syllabic delimiter). Statistical approaches such as IBM source-channel model from machine translation are often adopted to tackle transliteration problems among resource-rich languages such as English and Mandarin Chinese [2][3]. However, few studies have investigated transliteration approaches for under-resourced languages such as Vietnamese [4]. Given the limited corpus size of such under-resourced languages, performance of statistical transliteration approaches are often expected to be suboptimal.

In this work, we present a rule-based Vietnamese transliteration system, investigate its transliteration performance in comparison with classic statistical baselines using corpora of different Vietnamese dialects and various training data sizes, and evaluate how well different transliteration approaches perform in the context of ASR-based keyword search experiments. This work is part of a larger endeavor to build a unified transliteration framework for under-resourced Southeast Asian and Chinese languages which share a similar phonological structure.

2. Proposed transliteration Framework

2.1. Vietnamese phonology

2.1.1. Phonetic representation

The Vietnamese phonetic representation used are X-SAMPA symbols [5], while the English phonetic representation follow those in the CMU pronunciation dictionary [6]. All foreign words are converted to English phonetic representation.

2.1.2. Syllable structure

The English syllable structure is very rich, including possible structures such as CV, CVC, CCVC, CCCVC, where C represents consonant and V represents vowel. By contrast, the syllable structure for Vietnamese is simpler. Vietnamese is a monosyllabic language, where syllables are separated by white space in text. A word can be represented by one syllable or multiple syllables while one syllable may or may not have a meaning [7]. A Vietnamese syllable [8] is usually segmented as:

[C1][w]V[C2] + T,

where the "[]" specifies an optional segment, C1 is the syllable-initial consonant, w is the glide, V is the vowel nucleus (which is a vowel or vowel cluster), C2 is the syllable-ending consonant, and T is the tone.

2.1.3. Vowels

The nucleus of Vietnamese syllable is a vowel or vowel cluster. Below are some important phonological constraints on Vietnamese vowels [9]:

- All monophthong vowels except /a/ and /@/ can
 1. occur in any position of a syllable (i.e., initial, medial, final), or
 2. be an individual syllable in its own right.
- /a/ and /@/ can only occur in syllable-final positions.
- /E/ cannot be followed by /t/, /c/, or /ng/.

2.1.4. Consonants

There are a total of 20 Vietnamese consonants, none of which can occur medially in syllables. All consonants in Vietnamese

can occur in syllable-initial positions, except for /p/. However, /p/ occurs in initial position of many foreign words when they are adopted to Vietnamese [10].

2.1.5. Lexical Tones

The northern dialect is the standard variety (aka Ha Noi), which has six lexical tones: tone 1 (Level tone), tone 2 (Rising tone), tone 3 (Falling tone), tone 4 (Curve tone), tone 5 (Broken tone), and tone 6 (Drop tone). In other dialects, there are typically only five tones, where tone 4 and 5 are often merged [9].

2.2. Proposed transliteration algorithm

In the following we delineate our proposed algorithm for transliterating Vietnamese.

1. **Text-to-phoneme conversion:** The foreign word is converted to English phonetic representations using the t2p tool developed at CMU [11]. Despite the different origin language sources, we assume the foreign text is English. This assumption is based on: (1) A large portion of these foreign words are of English origin; (2) Vietnamese human transliterators often assume the foreign text follows English orthography and phonology, because they are most familiar with English as a 2nd language.
The phoneme sequence is used in conjunction with the original English grapheme sequence (text form) of the word for the following reasons: (1) When the correct phonemes of a foreign word are not obvious, the transliterator relies on the graphemic form of the foreign word to approximate the phonemes. (2) The CMU t2p tool targets text-to-phoneme conversion for North American English words, which could result in errors for non-English words. The graphemic form of the word is used to correct such errors.
2. **Syllable splitting:** Syllables are the basic units of the Vietnamese language. The same phoneme can have different forms in text depending on its position in the syllable. We enumerate the steps of syllable splitting below.
 - (a) **Identify vowel nuclei:** Vowels are used as the anchor point to specify each syllable.
 - (b) **Locate syllable-initial consonants:** All possible syllable-initial consonants (or consonant clusters) are located to maximize the number of syllables with initial consonants. This strategy is based on the fact that the majority of Vietnamese syllables start with a consonant.
 - (c) **Locate syllable-final consonants:** Syllable-final consonants (or consonant clusters) are located to complete the syllable if necessary.
3. **Foreign phoneme to Vietnamese phoneme mapping:** The mappings are determined by the Language Specific Peculiarities Document of OpenKWS13 and [9].
4. **Tone specification:** If the syllable ends with stop consonants /p, t, k/, tone 2 (rising) is assigned, else tone 1 (level) is assigned [9].
5. **Vietnamese syllables post-processing:** Vietnamese grammar dictates certain adjacent phonemes to be combined to form rhymes. For instance, adjacent vowels “a:” and “o” are combined to form the rhyme “a:U” [12].

3. Baseline models

We consider two classic transliteration algorithms. The transliteration problem can be formulated as follows: Given a string of tokens (e.g., letters, graphemes, phonemes) $\mathbf{f} = (f_1, f_2, f_m)$ from language F , the objective is to convert this input string into a string of tokens $\mathbf{e} = (e_1, e_2, e_n)$ from language E .

3.1. IBM source-channel model

The IBM source-channel model [13] is one of the most widely studied machine translation models and frequently adopted to solve transliteration problems (e.g., [3, 4]). The transliteration process can be formalized as

$$\mathbf{e}^* = \arg \max_{\mathbf{e}} p(\mathbf{e})p(\mathbf{f}|\mathbf{e}), \quad (1)$$

where the distribution of tokens $p(e)$ imposes a language constraint to the transliteration output, often estimated from a trigram language model; the $p(\mathbf{f}|\mathbf{e})$ is conditional probability of string $\mathbf{f}$ from language F given string $\mathbf{e}$ from language E , which can be estimated from forward backward algorithms used in hidden Markov models [14].

3.2. Joint source-channel model

The joint source-channel model for transliteration was introduced in [15]. Similar approaches have also been proposed for grapheme-to-phoneme conversion in [16]. In contrast to the IBM source-channel model, the joint source-channel model estimates the most optimal transliteration string $\mathbf{e}^*$ without decomposing the joint probability $p(\mathbf{e}, \mathbf{f})$:

$$\mathbf{e}^* = \arg \max_{\mathbf{e}} p(\mathbf{e}, \mathbf{f}). \quad (2)$$

The transliteration process can be seen as finding the alignment for sub-sequences of the input string $\mathbf{f}$ and the output string $\mathbf{e}$ [17]. Let there be K aligned transliteration pairs $\langle x, y \rangle_1, \dots, \langle x, y \rangle_K$, where x takes on sub-sequences of $\mathbf{e}$ and y takes on sub-sequences of $\mathbf{f}$. The joint probability $p(\mathbf{e}, \mathbf{f})$ can be estimated through an N-gram transliteration model, where the k -th alignment pair $\langle x, y \rangle_k$ is dependent on its N predecessor pairs [15]:

$$p(\mathbf{e}, \mathbf{f}) = \prod_{k=1}^K P(\langle x, y \rangle_k | \langle x, y \rangle_{k-N+1}, \dots, \langle x, y \rangle_{k-1})$$

4. Transliteration Experiment

In this section, we evaluate how well the proposed transliteration model converts foreign text to Vietnamese phonemes.

4.1. Experimental Setup

4.1.1. Corpus: NIST OpenKWS13 Evaluation data

The Vietnamese corpus is released from the IARPA Babel program [18] for the NIST OpenKWS13 Evaluation. The corpus contains dialects of four regions (Northern, Northern-central, Central, Southern). Since all words have a Northern pronunciation and it is the standard dialect, we only used Northern pronunciation of the foreign words. Foreign words were determined manually by a native Vietnamese speaker.

The data comes with a full language pack (FullLP), containing 100 hours of transcribed speech, and a limited language pack (LimitedLP), which is a 10 hour subset of FullLP. We further partitioned the data to intermediary language packs (IntLP1, IntLP2, IntLP3). The training, developmental, and test sets of each language pack are partitioned as shown in Table 1; they all share the same test set of 140 words.

Table 1: Partition of NIST OpenKWS13 Evaluation data

Set	LimitedLP	InterLP1	InterLP2	InterLP3	FullLP
Train	76	148	246	345	483
Dev	18	37	62	86	105
Test	140	140	140	140	140

Table 2: Transliteration performance on FullLP set.

Model	TER (%)	SyLER (%)	SER (%)
IBM source channel model	25.8	18.3	69.3
Joint source channel model	24.4	18.7	61.4
Proposed model	19.8	13.6	52.9

Table 3: Transliteration performance on LimitedLP set.

Model	TER (%)	SyLER (%)	SER (%)
IBM source channel model	55.3	74.3	94.7
Joint source channel model	65.6	47.0	97.1
Proposed model	19.8	13.6	52.9

4.1.2. Implementation details

The IBM Source-channel model was implemented using Moses statistical machine translation system [19]. The Joint source-channel model was implemented with Sequitur G2P [20].

4.1.3. Evaluation metrics

1) Token error rate (TER): Tokens include both phones and syllable delimiters.

2) Syllabic error rate (SyLER): for each lexicon, a set of valid syllables is created by extracting all distinct syllables in the training, developmental and test set. A syllable in the transliteration output is considered invalid if it does not exist in the set of valid syllables of the lexicon.

3) String error rate (SER): Any token error within a string leads to string error. TER and SER are computed using the NIST SCLITE tool [21].

4.2. Results

From Figure 1a, 1b, and 1c, we see that the proposed model consistently outperforms the statistical baselines for all three evaluation metrics, despite the corpus size. Note that the performance of the proposed model remained constant regardless of the training data size since it is not-statistical in nature.

Below we analyze in more detail the FullLP and LimitedLP conditions. Table 2 shows that the proposed system consistently outperforms the baselines over the FullLP set: it improves Joint source-channel model by 18.85% relative in TER, by 27.27% relative in SyLER, and by 13.84% in SER; it improves the IBM source channel model by 23.26% relative in TER, by 25.68% in SyLER, and by 23.67% relative in SER.

Table 3 shows that the proposed system consistently outperforms the baselines over the LimitedLP set: it improves Joint source-channel model by 69.82% relative in TER, by 71.06% in SyLER, and by 45.52% relative in SER; it improves the IBM source channel model by 64.20% relative in TER, by 81.70% relative in SyLER, and by 44.14% relative in SER.

4.3. Further analysis: data size vs. performance trade-off

To gain deeper insight into on how effective the proposed rule-based approach is compared to statistical methods, we analyzed corpus size and performance trade-off on a larger corpus.

4.3.1. HCMUS corpus

This corpus was kindly provided from researchers at Ho Chi Minh City University of Science (HCMUS) [4], and contains 4,256 pairs of foreign words and their corresponding transliterated Vietnamese words. The pronunciations in this corpus are from central and southern dialects in Vietnam. Sub-corpora sizes of 500, 1,000, 2,000, 3,000 and 4,000 were randomly chosen from the 4,256 pairs. Each sub-corpus was subsequently split randomly into three sets of the proportion 60%, 20%, 20% for training, development and testing, respectively.

4.3.2. Results

As shown in Figures 2a, 2b, and 2c, the proposed model performs consistently better than the baseline IBM Source-channel model for all corpus sizes, and in all three metrics. The proposed algorithm performs better than the Joint Source-channel model for corpus size up to 1,000. These analyses help us determine which models to use depending on the corpus size.

Syllabic error rate (SyLER) also indicates the advantage of the proposed transliteration model. The proposed model performs consistently better than both statistical model for all corpora sizes. We identify two main causes of syllabic error rate in the statistical models: (1) the statistical models cannot model the phonological rules to generate a valid Vietnamese syllable, and thus, generate syllables that violate phonological constraints, e.g. vowel constraints in section 2.1.3 (2) the statistical models treat syllabic delimiter as equivalent to other phones, and thus, generate syllables with invalid structure, e.g. there is no or more than one vowel nucleus between a pair of syllabic delimiter.

5. Keyword search experiment

Since transliteration is important for resolving OOV issues in spoken language technology applications such as automatic speech recognition (ASR) and keyword search, especially when language resources are limited, we also evaluate our transliteration model under a keyword search scheme.

5.1. Keyword search system

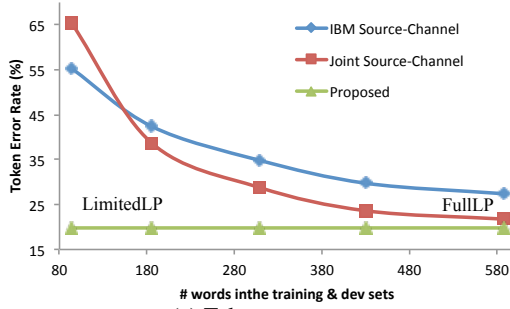
We employed a state-of-the-art ASR-based keyword search system (see Figure 3) developed at the Institute for Infocomm Research [23]: Perceptual linear prediction features and tonal features (the fundamental frequency estimate and fundamental frequency variation) were used to train a deep neural network acoustic model. The pronunciation dictionary was from the OpenKWS13 Evaluation full language pack with modifications: the pronunciations of the 140 foreign keywords described in Section 4.1.1. were replaced with transliterated pronunciations of the different transliteration models. The language model was a syllable-based bigram system. The ASR lattice indexing and search were done using Kaldi [24]. More details of the keyword search system can be found in [23]. The data partition is the same as in Section 4.1.1.

5.2. Evaluation metric: term-weighted value (TWV)

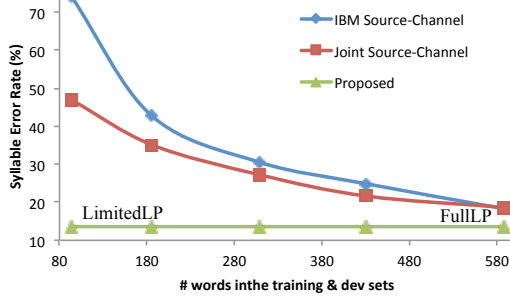
TWV is 1 minus the weighted sum of the term-weighted probability of miss detection $P_{miss}(\theta)$ and the term-weighted probability of false alarm $P_{FA}(\theta)$:

$$TWV(\theta) = 1 - [P_{miss}(\theta) + \beta P_{FA}(\theta)] \in (\infty, 1], \quad (3)$$

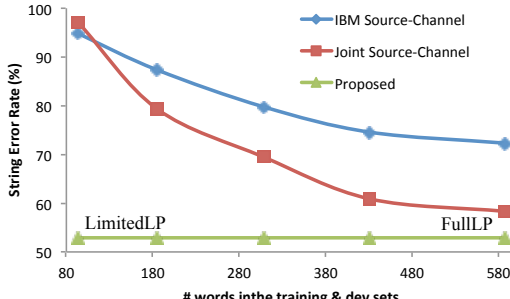
where θ is the decision threshold. The larger the TWV value, the better the performance. Actual term-weighted value



(a) Token error rate

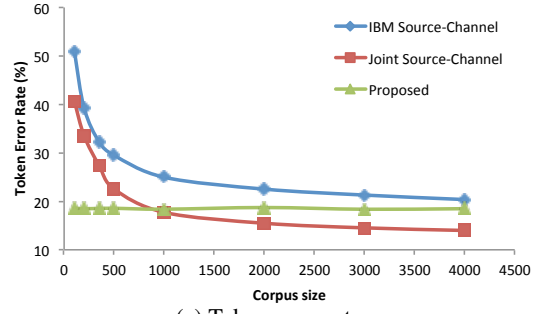


(b) Syllabic error rate

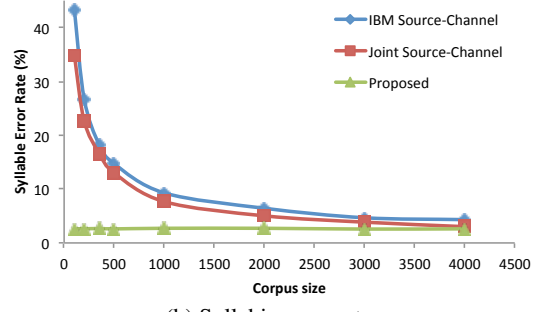


(c) String error rate

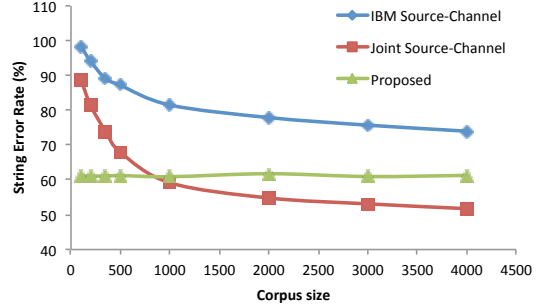
Figure 1: Performance of different transliteration models as a function of corpus size (NIST OpenKWS13 dataset).



(a) Token error rate



(b) Syllabic error rate



(c) String error rate

Figure 2: Performance of different transliteration models as a function of training data size (HCMUS corpus).

(ATWV) is the TWV of the chosen decision threshold, whereas maximum term-weighted value (MTWV) is the most optimal TWV among all decision thresholds.

5.3. Results

The keyword search results are shown in Table 4. In terms of ATWV, System S3 (proposed model) outperforms System S1 and System S2 by 36.76% relative (2.11% absolute) and 14.93 % relative (1.02% absolute), respectively. In terms of MTWV, System S3 (proposed model) outperforms System S1 and System S2 by 30.56 % relative (2.17% absolute) and 21.01% relative (1.61% absolute), respectively. Note that the comparison of MTWV serves as an upper-bound performance gap. Even if the baseline systems S1 and S2 used their best decision thresholds, their upper-bound performance would not have reached the ATWV of the proposed System S3.

6. Discussion

To the best of our knowledge, our work is the first Vietnamese rule-based transliteration framework. While the transliteration

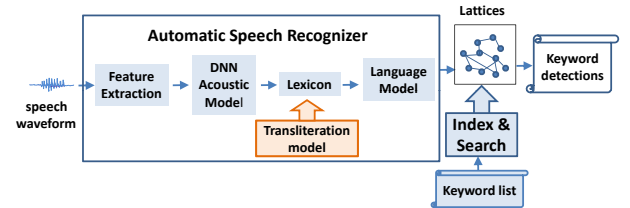


Figure 3: An ASR-based keyword search system.

Table 4: Keyword search system performance when adopting various transliteration models.

System	Transliteration model	ATWV	MTWV
S1	IBM source channel model	0.0574	0.0710
S2	Joint source channel model	0.0683	0.0766
S3	Proposed model	0.0785	0.0927

system presented is based on Vietnamese, we are currently extending this transliteration framework to other languages. We aim to develop a common rule-based transliteration tool for Southeast Asian and Chinese languages which share a similar phonological and tonal structure.

7. References

- [1] Kevin Knight and Jonathan Graehl, "Machine transliteration," *Computational Linguistics*, vol. 24, no. 4, pp. 599–612, Dec. 1998.
- [2] Wei Gao, Kam-Fai Wong, and Wai Lam, "Phoneme-based transliteration of foreign names for OOV problem," in *Proceedings of the First International Joint Conference on Natural Language Processing*, Berlin, Heidelberg, 2005, IJCNLP'04, pp. 110–119, Springer-Verlag.
- [3] Paola Virga and Sanjeev Khudanpur, "Transliteration of proper names in cross-language applications," in *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval*, New York, NY, USA, 2003, SIGIR '03, pp. 365–366, ACM.
- [4] Nam X. Cao, Nhut M. Pham, and Quan H. Vu, "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*, New York, NY, USA, 2010, SoICT '10, pp. 59–63, ACM.
- [5] John C. Wells, "Computer-coding the ipa: a proposed extension of sampa: <http://www.phon.ucl.ac.uk/home/sampa/x-sampa.htm>," 2001.
- [6] Carnegie Mellon University, "Cmu pronunciation dictionary: <http://www.speech.cs.cmu.edu/cgi-bin/cmudict>," .
- [7] Hong Quang Nguyen, Pascal Nocera, Eric Castelli, and Van Loan Trinh, "A novel approach in continuous speech recognition for vietnamese, an isolating tonal language," in *INTERSPEECH*. 2008, pp. 1149–1152, ISCA.
- [8] Dinh Hoa Nguyen, *Vietnamese*, John Benjamins Publishing Co., Amsterdam, Netherlands, 1997.
- [9] Thi-Quynh-Hoa Hoang, "A phonological contrastive study of Vietnamese and English," MA Thesis, Texas Technology College, 1965.
- [10] Giang M. Tang, "Cross-linguistic analysis of vietnamese and english with implications for vietnamese language acquisition and maintenance in the united states," *Journal of Southeast Asian American Education & Advancement*, vol. 2, 2007.
- [11] Kevin Lenzo, "t2p: Text-to-phoneme converter builder. retrieved from carnegie mellon university: <http://www.cs.cmu.edu/afs/cs.cmu.edu/user/lenzo/html/areas/t2p>," 1998, December 28.
- [12] Alexis Michaud, "Final consonants and glottalization: new perspectives from hanoi vietnamese," *Phonetica*, vol. 61, no. 2-3, pp. 119–146, 2004.
- [13] Peter F. Brown, Vincent J. Della Pietra, Stephen A. Della Pietra, and Robert L. Mercer, "The mathematics of statistical machine translation: Parameter estimation," *Comput. Linguist.*, vol. 19, no. 2, pp. 263–311, June 1993.
- [14] Lawrence Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.
- [15] Haizhou Li, Min Z, and Jian Su, "A joint source-channel model for machine transliteration," in *Proceedings of the 42Nd Annual Meeting on Association for Computational Linguistics*, Stroudsburg, PA, USA, 2004, ACL '04, Association for Computational Linguistics.
- [16] Maximilian Bisani and Hermann Ney, "Joint-sequence models for grapheme-to-phoneme conversion," *Speech Communication*, vol. 50, no. 5, pp. 434 – 451, 2008.
- [17] Vladimir Pervouchine, Haizhou Li, and Bo Lin, "Transliteration alignment," in *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1-Volume 1*. Association for Computational Linguistics, 2009, pp. 136–144.
- [18] NIST, "Nist open keyword search 2013 (openkws13) evaluation: <http://www.nist.gov/itl/iad/mig/openkws13.cfm>," .
- [19] "Moses statistical machine translation system: <http://www.statmt.org/moses>," .
- [20] Maximilian Bisani, "Sequitur g2p: A trainable grapheme-to-phoneme converter: <http://www-i6.informatik.rwth-aachen.de/web/software/g2p.html>," .
- [21] NIST, "Nist tools: <http://www.itl.nist.gov/iad/mig/tools/>," .
- [22] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu, "Bleu: A method for automatic evaluation of machine translation," in *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, Stroudsburg, PA, USA, 2002, ACL '02, pp. 311–318, Association for Computational Linguistics.
- [23] Nancy F. Chen, Sunil Sivadas, Boon Pang Lim, Hoang Gia Ngo, Haihua Xu, Van Tung Pham, Bin Ma, and Haizhou Li, "Strategies for vietnamese keyword search," in *ICASSP*, 2014.
- [24] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, et al., "The kaldi speech recognition toolkit," in *Proc. of IEEE ASRU*, 2011.