Word Transduction for Addressing the OOV Problem in Machine Translation for Similar Resource-Scarce Languages

Shashikant Sharma and Anil Kumar Singh
IIT (BHU), Varanasi, India
{shashikant.sharma.cse12, aksingh.cse}@iitbhu.ac.in

Abstract

Similar languages have a large number of cognate words which can be exploited to deal with Out-Of-Vocabulary (OOV) words problem. This problem is especially severe for resource-scarce languages. We propose a method for 'word transduction' for addressing this problem. We take advantage of the fact that, although it is difficult to prepare sentence aligned parallel corpus for such languages, it is much easier to prepare 'parallel' list of word pairs which are cognates and have similar pronunciations. We can try to learn pronunciations (or orthographic representations) of OOV words from such a parallel list. This could be done by using phrase-based machine translation (PBMT). We show that, for small amount of data, a model based on weighted rewrite rules for phoneme chunks outperforms a PBMT-based approach. An additional point that we make is that word transduction can also be used to borrow words from another similar language and adapt them to the phonology of the target language.

1 Introduction

Current research in the field of Automatic Speech Recognition (ASR) and Machine Translation (MT) tends to focus on the language pairs that have a large amount of data available. This is because the quality of these systems is dependent on the amount and quality of the training data used. As a result, many such systems between (relatively) resource-rich languages, such as English, Hindi, and Urdu are available. However in a country like India, there are 122 major languages and over 1599 other languages spoken by different communities[1]. Most of these languages are resource-scarce and therefore very little or no work has been done on these languages. Language is one of the major factors responsible for digital divide between urban and rural areas due to prevalence of information technology in urban areas (Dubey and Devanand, 2013). Therefore, removing this language barrier is crucial to the growth of the society as well as for bridging the digital divide.

Hindi has several 'dialects' (often called sub-languages) spread over the entire Hindi speaking region commonly known as the Hindi-Belt. Bhojpuri is one of the seven Hindi sub-languages (other six include Awadhi, Braj, Haryanvi, Bundeli, Bagheli and Kannauji) (Mishra and Bali, 2011). Although the designation of Bhojpuri as a language or simply as a dialect of Hindi is a topic of debate[2], it is closely related to Hindi and borrows many words from Hindi, either directly or with some phonological (and thus, orthographic) changes. Besides, both use the Devanagari script for all official purposes, which (like major Indian languages, Indo-Aryan and Dravidian) has evolved from the ancient Brahmi script (Sproat, 2002; Sproat, 2003). In spite of having more than 33 million speakers[3], Bhojpuri is a resource-scarce language due to lack of resources such as machine readable dictionaries, WordNet and any standard parallel corpus, which makes the development of machine translation (MT) systems very challenging. For the same reason, Statistical Machine Translation is not feasible as it requires a

[1] http://www.censusindia.gov.in/Census_Data_2001/Census_Data_Online/Language/gen_note.html

[2] http://ncictt.com/index.php/articles/42-bhojpuri-a-dialect-of-hindi

[3] http://www.censusindia.gov.in/Census_Data_2001/Census_Data_Online/Language/Statement1.aspx

Proceedings of the 13th International Conference on Finite State Methods and Natural Language Processing, pages 56–63,
Umeå, Sweden, 4–6 September 2017.
https://doi.org/10.18653/v1/W17-4007

large parallel corpus. Alternatively, direct MT is more suitable for closely related languages (Hajič et al., 2000).

Therefore, preparation of sufficient amount of data for NLP tools like MT systems seems very difficult in the near future. Any such system, due to lack of resources, faces low-coverage issues due to the presence of unknown (Out-Of-Vocabulary or OOV) words.

To address this issue, we propose a 'word transduction' approach, which can show noticeable improvement in inter-dialectal translation by transducing OOV words. We define the term word transduction as the conversion of words from one source language to another closely related target language such that pronunciation and meaning are similar. This can have two aspects. One is cognate generation, while the other is adapting borrowed words from the source language to the target language such that it matches the phonology of the target language. In other words, word transduction can be seen as transliteration (Denoual and Lepage, 2006; Finch and Sumita, 2009) *in the same script* to incorporate phonological changes between a pair of closely related languages.

Since the problem of machine transliteration can also be viewed as the process of machine translation at the character level, we have used the popular phrase-based SMT (Statistical Machine Translation) system Moses between Hindi-Bhojpuri word pairs as the baseline system. SMT requires a bilingual parallel training data, a language model (LM), a translation model (TM) and a decoder. This method uses mapping of small text chunks (called 'phrases') without the utilization of any explicit linguistic information (such as morphological, syntactic, or semantic). That is, it considers only the surface form of words to create a phrase table. Such additional information can be incorporated in the form of 'factors', along with the words or characters (in case of transliteration) to improve the accuracy of standard SMT.

Surface form, along with these factors, creates factored representation of each word (Koehn et al., 2007; Koehn and Hoang, 2007). For factored SMT, we augmented each letter of Devanagari with their phonetic features (described in the next section) to create its factored representation. This factored Statistical Machine Translation at character level performed better than our baseline SMT system.

The representation of speech using a sequence of phonetic symbols is defined as transcription. Hindi has a phonetic writing system, i.e., there is very little distinction between its transcription and pronunciation. Therefore, it is reasonable to assume that words in Hindi and Bhojpuri are spelled or transcribed in the same way as they are pronounced (Choudhury, 2003). This property makes the mapping of Devanagari letters to International Phonetic Alphabet (IPA) symbols very easy. IPA is organized in such a way that each symbol on a chart can be visualized as a hierarchical structure of features (Peter Ladefoged, 1988). It is possible to decomposes letters in the IPA representation into the building block of sounds (features). We have used the IPA representation of both source (Hindi) and target (Bhojpuri) language. For instance, क is a single Devnagari alphabet but is equivalent to क् + अ, i.e., क has the inherent vowel अ which is easily reflected in its IPA representation(/kə/).

The major contributions of this paper are:

- We propose a phoneme chunk based method for word transduction which transduces words of Hindi to its closely related language Bhojpuri. Using the method described in this paper, we try to predict Bhojpuri pronunciation from its corresponding Hindi word. This method is adapted from extensively reported earlier work on similar problems.

- We also show that proposed phoneme chunk based method for word transduction performs better than the standard Statistical Machine Translation as well as factored Statistical Machine Translation (Koehn and Hoang, 2007) when applied on the same dataset using the Moses decoder (Koehn et al., 2007).

2 Related Work

In a parallel corpus of a language and its dialect (or closely related language), words can be categorized into two categories based on their pronunciation or orthographic form:

- Word pairs having entirely different pronunciations (and hence orthographic forms) in the two varieties. For example, रउआ (rauaa) in Bhojpuri means आप (aap, you-honorific) in Hindi. This type of word pairs share almost no or very little phonetic and orthographic similarity. Since our model utilizes

phonetic transition between two closely related languages, this type of word-pairs are not suitable for our model.

- Words having similar pronunciations (and hence their orthographic forms). A phonemic study of Hindi and Bundeli (Acharya, 2015), mainly focusing on the prosodic features and the syllabic patterns of these two languages concluded that the borrowing of words from Hindi to Bundeli generally follows certain rules. For instance if a word in Hindi starts with य [ya], it is replaced by ज [ja] in its Bundeli equivalent as यजमान [yajamaan] becomes जजमान [jajamaan], यमुना [yamunaa] becomes जमुना [jamunaa] etc. This category of word-pairs is our main motivation behind the work described in this paper. Our goal was to build a system which takes a word as input in one language and returns its equivalent in some other language which is closely related to source language by using the phoneme to phoneme conversion. The next section will describe the steps of the proposed method.

Koo (2011) proposed a model using weighted finite state transducers (WFST) to implement phoneme rewrite rules for English to Korean. This finite state model was applied to predict how English words and named entities are pronounced in Korean by a native speaker of Korean. Initially the model keeps one or more rewrite rules for every phoneme in English, then each rewrite rule specializing in a given English phoneme is weighted according to the probability with which the rule applies. Each rewrite rule is defined as:

$$\phi \rightarrow \varphi / \lambda\_\rho$$

i.e., rewrite ϕ as φ when preceded by λ and followed by ρ. ϕ is an English phoneme, φ is a phoneme of Korean. In other words, these rules, consisting of three basic operations, can be implemented as the union of three WFSTs, i.e., deletion, substitution and substitution plus insertion.

Koskenniemi (2013) modelled correspondences between two historically related languages (Finnish and Estonian, derived from Proto Balto-Finnic or PBF) using finite state transducers (FST). Using general linguistic knowledge, Finnish and Estonian forms were aligned letter by letter with each other and these aligned words, known as Aligned Finnish-Estonian (AFE), were used as a substitute for the proto-language. AFE, having more symbols than the normal set of phonemes (as in PBF), was applied to produce Finnish, Estonian and PBF directly and unambiguously.

Singh (2006) formulated a phonetic model to represent relations between the sounds of Indian languages and the letters or 'akshars' (orthographic syllables). It included phonetic features (Clements, 1985) mapped to each letter as well as a computational model to calculate the orthographic and phonetic distance between given pair of akshars, letters, words or strings. The phonetic features described were mainly the ones considered in modern phonetics, as well as some orthographic features specific to Indian language scripts. The distance measure was based on the fact that phonetic features differentiate two sounds (or akshars representing them) in a cascaded or hierarchical way. The features that we have used for factored SMT are selected from the higher levels only, since the Moses decoder only allows four factors at most. These features, along with their possible values, are listed in Table 1.

Features	**Possible Values**
Type	Unused, Vowel modifier, Nukta, Halant, Vowel, Consonant, Number, Punctuation
Height (vowels)	Front, Mid, Back
Sthaan (place)	Dvayoshthya, Dantoshthya, Dantya, Varstya, Talavya, Murdhanya, Komal-Talavya, Jivhaa-Muliya, Svaryantra-mukhi
Prayatna (manner)	Sparsha, Nasikya, Parshvika, Prakampi, Sangharshi, Ardh-Svar

Table 1: Phonetic features and their possible values

3 Proposed Model

Previous studies have shown that we can use weighted finite state transducers (WFST) to generate rewrite rules for translation between closely related languages (Koskenniemi, 2013) as well as to generate phonological rules (Koo, 2011; Gildea

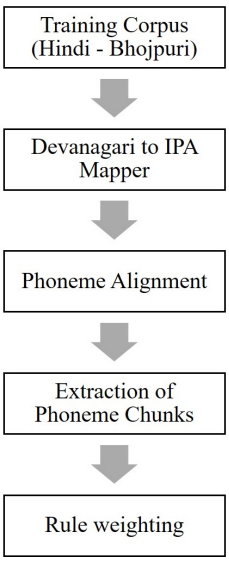

Figure 1: Structure of the Training Model

and Jurafsky, 1995) for word pronunciation. However, when the implementation of a method similar to Koo (2011) was applied on Hindi-Bhojpuri parallel list, it yielded poor results. This was due to the fact that model proposed by Koo considered only single phonemes of the source language (English) while formulating the rewrite rules. Our proposed method, instead of considering single phonemes, considers rewrite rules of all possible chunks and weights them according to the frequency of occurrence in the dataset. Furthermore, general rewrite rules were defined using regular expressions to post-process the ranked output from the trained model.

3.1 Training the Model

The structure of the training model is shown in the figure 1. Each of these steps is described in detail below:

1. **Devanagari to International Phonetic Alphabet Converter.** Using the assumption that Devanagari alphabets have the same pronunciation and transcription, we can map each Devanagari letter to their equivalent IPA using a freely available mapping[4]. This mapping is mostly one-to-one as each Devanagari letter has a unique mapping to IPA and vice-versa. Phoneme length of a word is the count of these IPA units. For example [b$^{\text{ɦ}}$] for भ् is a single phoneme. Note that phoneme length and character length are two different terms which will be used later during alignment of phonemes. Consider the following examples:

   Example 1: लगे (/ləgeː/) has $character\ length = 5$ and $phoneme\ length = 4$
   Example 2: डगमगाना (/ɖəgəməgaːnaː/) has $character\ length = 12$ and $phoneme\ length = 10$

2. **Alignment of the source and the target words for rule extraction.** This is the key step for our model. In this step, we align phonemes of word pairs in such a way that it has minimum phonetic distance (Singh, 2006). In our case, Hindi is the source language and Bhojpuri is the target language. These word pairs may have different phoneme lengths and, therefore, three types of rewrite rules are possible: *Substitution, Deletion and Insertion*. Put differently, a rewrite rule in this case defines how a Hindi phoneme should be edited via deletion, substitution, or insertion depending on which phoneme appears on both sides. For example, डगमगाना (/ɖəgəməgaːnaː/) → डगमगाइल (/ɖəgəməgaːilə/) have one phoneme insertion and two phoneme substitutions.

   We redefine an IPA representation of Hindi-Bhojpuri word pairs by inserting a placeholder ε until phoneme length of both source and target becomes equal and have minimum possible phonetic distance. For example, /ɖəgəməgaːnaː → /ɖəgəməgaːilə/ after alignment becomes /ɖəgəməgaːɛnaː/ → /ɖəgəməgaːilə/. Since Hindi-Bhojpuri word pairs now have equal phoneme length, only one type of rewrite rule, i.e., substitution is required.

3. **Extraction of Phoneme Chunks.** From aligned training data, we extract phoneme chunks (phoneme n-grams). We enumerate all possible phoneme substrings of the Hindi word for a given Hindi-Bhojpuri aligned pair. Since phoneme length is the same, a phoneme chunk of Hindi will directly map to a phoneme chunk of Bhojpuri of the same length (see figure 2). For example, after alignment with its Bhojpuri translation, परेम (/pəreːmə/), प्रेम (/preːmə/) will have the IPA representation /pɛreːmə/ and the constituent

[4] https://en.wikipedia.org/wiki/Help:IPA_for_Hindi_and_Urdu

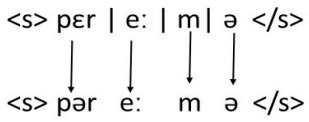

Figure 2: Mapping Hindi phoneme chunks to Bhojpuri phoneme chunks.

phoneme chunks can be generated as shown in Table 2.

Hindi	Bhojpuri
<s> p | ɛre:mə <s>	<s> p | əre:mə <s>
<s> pɛ | re:mə <s>	<s> pə | re:mə <s>
<s> pɛr | e:mə <s>	<s> pɛr | e:mə <s>
<s> pɛre: | mə <s>	<s> pɛre: | mə <s>
<s> pɛre:m | ə <s>	<s> pəre:m | ə <s>
<s> p | ɛ | re:mə <s>	<s> p | ɛ | re:mə <s>
<s> ..... <s>	<s> ..... <s>

Table 2: Some entries of the aligned phoneme chunks for word प्रेम (each phoneme chunk is separated by "|", and <s>is a word boundary marker)

4. **Rule weighting.** Each phoneme chunk can be transduced to phonemes of a Bhojpuri word of the same length as shown in figure 3. Therefore, each rewrite rules derived will be of the form:

$$\alpha \rightarrow \beta$$

where α is a phoneme chunk of Hindi and β is a Bhojpuri phoneme chunk of the same length. Rule derivation process after alignment consists of finding the probability of chunk translation and weighting each translation based on its weight W, defined as:

$$W(\alpha \rightarrow \beta) = (p(\alpha \rightarrow \beta))^2 * plen(\alpha)$$

where $plen(\alpha)$ is the phoneme length of α and $p(\alpha \rightarrow \beta)$ is the probability of translation α to β, calculated as:

$$p(\alpha \rightarrow \beta) = \frac{C(\alpha \rightarrow \beta)}{C(\alpha)}$$

$C(\alpha \rightarrow \beta)$ means the frequency of α translated to β in aligned phoneme chunks of training data and $C(\alpha)$ means a count of all the occurrences of α in aligned phoneme chunks of training data. Probability p was considered only if $p \geq 0.50$.

3.2 Estimating Bhojpuri Pronunciation

Estimating Bhojpuri pronunciation consists of two steps. Using weighted Hindi phoneme chunks, we first assign a rank to each possible translation, then we treat the phonemic representation of the highest ranked word from this output as an input to the general rewrite rule system. Put differently, as explained earlier, first all possible phoneme chunks are enumerated for the Hindi word whose Bhojpuri pronunciation is to be estimated, then for each row in aligned phoneme chunks (see Table 2), each of the phoneme chunks are tranduced to Bhojpuri and their weights are aggregated to calculate the rank of the transduced output. Highest ranked output is then passed as an input to the general rewrite rule system, which relies on the linguistic knowledge about the Bhojpuri language. This system consists of mapping of Hindi phonemes to Bhojpuri using regular expressions, and some of its rewrite rules are given below:
- $kʂ \rightarrow c^h$
- $ɳ \rightarrow n$
- $ɕ \rightarrow s$
- $ʋ \rightarrow b$
- $ːrj \rightarrow j$
- $<s>j \rightarrow <s>ʝ$ (<s> is a boundary marker for the start of a word)
- $ʂ \rightarrow s$
- $ʊθ \rightarrow ʊəθ$ (both $ʊ, θ$ are phonemic equivalent of a consonant)

4 Experiments

Experiments were performed from two points of view: the accuracy test and the phonetic distance comparison.

4.1 Dataset

The proposed model was trained and tested using a dataset consisting of 4220 Hindi-Bhojpuri word pronunciation pairs chosen from a lexicon compiled by language experts. The 4220 pairs were randomly split into a training set and a test set in a three-to-one ratio. The model was developed on 3165 pronunciation pairs and predicted the Bhojpuri pronunciation of Hindi words in the remaining 1055 pairs. Dataset consisted of word pairs of Type 2 words (described in Section 2). A sample of the corpus is shown in Table 3.

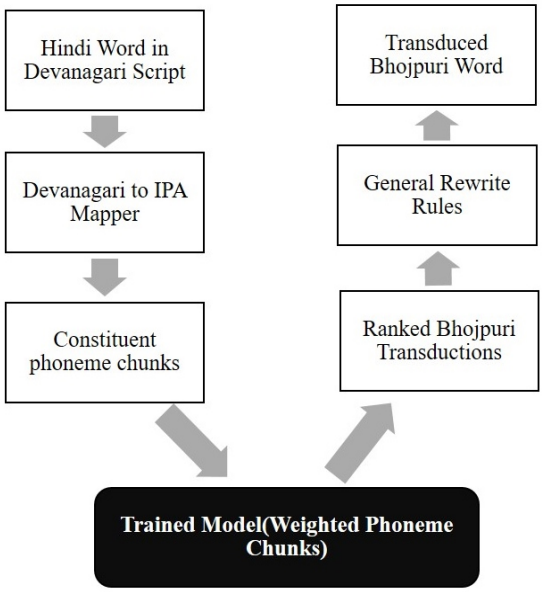

Figure 3: Estimating Bhojpuri Pronunciation from a Hindi word.

Hindi	Bhojpuri
किस्मत (kismata)	किसमत (kisamata)
ढिंढोरा (dhindhoraa)	ढिनढोरा (dhinadhoraa)
सताता (sataataa)	सतावल (sataavala)
विकसित (vikasita)	बिकसित (bikasita)
स्कूल (skUla)	इसकूल (iskUla)
कौआ (kau-aa)	कउआ (ka-u-aa)

Table 3: Sample word pairs from dataset (Roman transliteration in parenthesis)

4.2 Statistical Machine Translation

For this work, a bilingual Hindi-Bhojpuri Machine Translation Model has been used as the baseline by using the Moses decoder. Moses requires a parallel corpus (e.g. Hindi and Bhojpuri) that is used for training the system. For this task, each letter from the parallel corpus (parallel word list) of Hindi-Bhojpuri language pair was treated as if it was a word of a sentence and each word was treated as a single sentence. In other words, machine translation was performed at the character level. The publicly available tool GIZA++ was used to align the letters (Och and Ney, 2003). IRSTLM (Federico et al., 2008) was used to create the language model, which computes the probability of target language sentences (words in our case). Language model was prepared using the 19532 words, compiled from a Bhojpuri newspaper[5] and (a very limited) Hindi-Bhojpuri parallel corpora.

4.3 Factored Statistical Machine Translation

Moses (Koehn et al., 2007) provides framework for statistical translation models that easily integrates additional linguistic informations as factors. For the purpose of word transduction, each Devanagari letter was provided with its phonetic features to create its factored representation. As phonetic features differentiate between two sounds in a cascaded or hierarchical way, features were selected based on the level of hierarchy. Since the hard limit of factors in Moses is 3, we considered two different sets of phonetic features - the first set (we name it FSMT1) had features named Type, Height and Prayatna (manner), and the second one (we name is FSMT2) had Type, Height and Sthaan (place). This factored parallel corpus was then used to train the translation model using the SMT tools (Moses decoder, GIZA++ and IRSTLM).

Method	Word Accuracy (WA)
SMT	53.022%
FSMT1	54.746%
FSMT2	54.989%
Proposed Method	64.411%

Table 4: Word Accuracy Test

4.4 Evaluation Measures

Accuracy was measured by the percentage of the number of correctly transduced words divided by total number of generated transductions. We term it as word accuracy (WA). We define one more measure, called normalised phonetic distance (NPD) that measures the phonetic distance between a correct word and a generated word.

$$WA = \frac{Number\ of\ correct\ translation}{Total\ number\ of transduced\ words}$$

$$NPD(T,B) = \frac{PD(T,B) - PD_{min}}{PD_{max} - PD_{min}}$$

where $PD(T,B)$ is phonetic distance between transduced output T, and correct transduction B, computed using phonetic model as described by Singh (2006), PD_{min}, PD_{max} are minimum and maximum phonetic distance between transduced

[5] http://tatkakhabar.com/

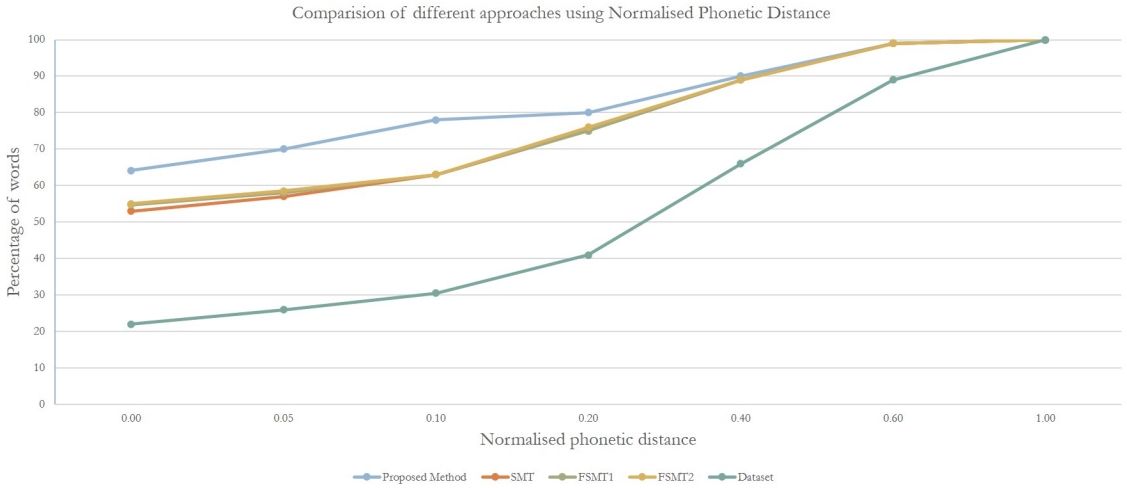

Figure 4: NPD comparison between proposed method and SMT on the same training and test dataset.

output and correct translation in the test corpus, respectively.

4.5 Accuracy Test

We compare our results to word accuracy of SMT and factored SMT in Table 4, which concludes that factored SMT with phonetic features as the factors performs better than standard SMT. However our phoneme chunk based method performs better than the other methods. This could be because we do not have enough data for SMT, which is the common scenario for resource-scarce languages.

4.6 BLEU Score

The BLEU score (Papineni et al., 2002) (which is one of the most popular measures for machine translation) for all the methods is summarised in Table 5. Here also our method significantly outperforms other methods for our language-pair.

Method	BLEU Score
SMT	75.05
FSMT1	75.92
FSMT2	76.18
Proposed Method	79.82

Table 5: BLEU score comparison

4.7 Normalised Phonetic Distance Test

We compare two methods also on the basis of Normalised Phonetic Distance (NPD). This test has physical significance in terms of pronunciation difference between generated output and the correct translation. The higher the value of accuracy for a given NPD, the closer the pronunciation of the transduced word and the correct transduction. We evaluated normalised phonetic distance (NPD) for five different word pairs:

1. Hindi-Bhojpuri word pairs of the test corpus (shown by the curve $Dataset$ in Figure 4)
2. Generated output by the SMT technique and its correct Bhojpuri output (shown by the curve SMT in Figure 4)
3. Generated output by the FSMT1 technique and its correct Bhojpuri output (shown by the curve $FSMT1$ in Figure 4)
4. Generated output by the FSMT2 technique and its correct Bhojpuri output (shown by the curve $FSMT2$ in Figure 4)
5. Generated output by the proposed technique and its correct Bhojpuri output (shown by the curve $Proposed\ Method$ in Figure 4)

From the comparison we can conclude that proposed method has better performance than SMT in reducing the pronunciation difference for the data size that we have.

5 Conclusion

We proposed an approach ('word transduction') for addressing the OOV problem for resource-scarce similar languages, of which one is more resource-scarce. Word transduction is aimed at guessing the pronunciation or the orthographic form of the target word, given the source word.

We learn to do this from a parallel list of cognate words. The approach can also be useful for adapting borrowed words to the phonology of the target language. We showed that a weighted rewrite rule-based method on phoneme chunks significantly outperforms a method based on factored phrase-based machine translation for this purpose for such language pairs. For future work, we plan to improve the implementation to make it faster and also to prepare more data so that the transducer can become practically useful, e.g. in an MT system. We are also trying Deep Learning methods for comparison. We plan to extend the dataset we have used and to release it for further work.

References

Ankita Acharya, 2015. *Contrastive Study of Bundeli and Hindi Pronunciation*. regICON-2015: Regional Symposium on Natural Language Processing, Varanasi.

Monojit Choudhury. 2003. Rule-based grapheme to phoneme mapping for hindi speech synthesis. In *90th Indian Science Congress of the International Speech Communication Association (ISCA), Bangalore, India*.

George N Clements. 1985. The geometry of phonological features. *Phonology*, 2(01):225–252.

Etienne Denoual and Yves Lepage. 2006. The character as an appropriate unit of processing for non-segmenting languages. In *NLP Annual Meeting*, pages 731–734.

Preeti Devanand Dubey and Devanand. 2013. Machine translation system for hindi-dogri language pair. In *Machine Intelligence and Research Advancement (ICMIRA), 2013 International Conference on*, pages 422–425. IEEE.

Marcello Federico, Nicola Bertoldi, and Mauro Cettolo. 2008. Irstlm: an open source toolkit for handling large scale language models. In *Interspeech*, pages 1618–1621.

Andrew Finch and Eiichiro Sumita. 2009. Transliteration by bidirectional statistical machine translation. In *Proceedings of the 2009 Named Entities Workshop: Shared Task on Transliteration*, pages 52–56. Association for Computational Linguistics.

Daniel Gildea and Daniel Jurafsky. 1995. Automatic induction of finite state transducers for simple phonological rules. In *Proceedings of the 33rd annual meeting on Association for Computational Linguistics*, pages 9–15. Association for Computational Linguistics.

Jan Hajič, Jan Hric, and Vladislav Kuboň. 2000. Machine translation of very close languages. In *Proceedings of the sixth conference on Applied natural language processing*, pages 7–12. Association for Computational Linguistics.

Philipp Koehn and Hieu Hoang. 2007. Factored translation models. In *EMNLP-CoNLL*, pages 868–876.

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.

Hahn Koo. 2011. A weighted finite-state transducer implementation of phoneme rewrite rules for english to korean pronunciation conversion. *Procedia-Social and Behavioral Sciences*, 27:202–208.

Kimmo Koskenniemi. 2013. Finite-state relations between two historically closely related languages. In *Proceedings of the workshop on computational historical linguistics at NODALIDA 2013; May 22-24; 2013; Oslo; Norway. NEALT Proceedings Series 18*, number 087, pages 53–53. Linköping University Electronic Press.

Diwakar Mishra and Kalika Bali. 2011. A comparative phonological study of the dialects of hindi. In *Proceedings of International Congress of Phonetic Sciences XVII*, pages 1390–1393.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

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.

Morris Halle Peter Ladefoged. 1988. Some major features of the international phonetic alphabet. *Language*, 64(3):577–582.

Anil Kumar Singh. 2006. A computational phonetic model for indian language scripts. In *Constraints on Spelling Changes: Fifth International Workshop on Writing Systems*. Nijmegen, The Netherlands.

Richard Sproat. 2002. Brahmi scripts. In *Constraints on Spelling Changes: Fifth International Workshop on Writing Systems, Nijmegen, The Netherlands*.

Richard Sproat. 2003. A formal computational analysis of indic scripts. In *International symposium on indic scripts: past and future, Tokyo*.