Finite-state Scriptural Translation

Muhammad Ghulam Abbas Malik, Christian Boitet, Pushpak Bhattacharyya

Finite-state Scriptural Translation

M. G. Abbas Malik **Christian Boitet** **Pushpak Bhattacharyya**
GETALP – LIG (Grenoble Informatics Lab.)
University of Grenoble IIT Bombay
Abbas.Malik Christian.Boitet@imag.fr pb@iitb.ac.in

Abstract

We use robust and fast Finite-State Machines (FSMs) to solve scriptural translation problems. We describe a *phonetico-morphotactic pivot* UIT (universal intermediate transcription), based on the common phonetic repository of Indo-Pak languages. It is also extendable to other language groups. We describe a *finite-state scriptural translation model* based on finite-state transducers and UIT. We report its performance on Hindi, Urdu, Punjabi and Seraiki corpora. For evaluation, we design two classification scales based on the word and sentence accuracies for translation system classifications. We also show that subjective evaluations are vital for real life usage of a translation system in addition to objective evaluations.

1 Introduction

Transliteration refers to phonetic translation across two languages with different writing systems, such as Arabic to English (Arbabi *et al.*, 1994; Stall and Knight, 1998; Al-Onaizan and Knight, 2002; AbdulJaleel and Larkey, 2003). Most prior work on transliteration has been done for MT of English, Arabic, Japanese, Chinese, Korean, etc., for CLIR (Lee and Choi., 1998; Jeong *et al.*, 1999; Fujii and Ishikawa, 2001; Sakai *et al.*, 2002; Pirkola *et al.*, 2003; Virga and Khudanpur, 2003; Yan *et al.*, 2003), and for the development of multilingual resources (Kang and Choi, 2000; Yan, Gregory et al., 2003).

The terms transliteration and transcription are often used as generic terms for various processes like transliteration, transcription, romanization, transcribing and technography (Halpern, 2002). In general, the speech processing community uses the term transcription to denote a process of conversion from the script or writing system to the sound (phonetic representation). For example, the transcription of the word "love" in the International Phonetic Alphabet (IPA) is [ləv]. While the text processing community uses the term transliteration and defines it as a process of converting a word written in one writing system into another writing system while preserving the sound of the original word (Al-Onaizan and Knight, 2002; AbdulJaleel and Larkey, 2003). More precisely, the text processing community defines the term transliteration as two transcription processes "source script to sound transcription" and "sound to target script transcription" and sometimes as one transcription process "source script to target script transcription".

We propose a new term *Scriptural Translation* for this combined process. Scriptural translation is a process of transcribing a word written in the source language script into the target language script by preserving its articulation in the original language in such a way that the native speaker of the target language can produce the original pronunciation.

FSMs have been successfully used in various domains of Computational Linguistics and Natural Language Processing (NLP). The successful use of FSMs have already been shown in various fields of computational linguistics (Mohri, 1997; Roche and Schabes, 1997; Knight and Al-Onaizan, 1998). Their practical and advantageous features make them very strong candidates to be used for solving *scriptural translation problems*.

First, we describe scriptural translation and identify its problems that fall under weak translation problems. Then, we analyze various challenges for solving weak scriptural translation problems. We describe our finite-state scriptural translation model and report our results on Indo-Pak languages.

Coling 2010: Poster Volume, pages 791–800,
Beijing, August 2010

2 Scriptural Translation – a weak translation problem

A weak translation problem is a translation problem in which the number of possible valid translations, say *N*, is either very small, less than 5, or almost always 1.

Scriptural Translation is a sub-problem of general translation and almost always a *weak translation problem*. For example, French-IPA and Hindi-Urdu scriptural translation problems are weak translation problems due to their small number of valid translations. On the other hand, Japanese-English and French-Chinese scriptural translation problems are not weak.

Scriptural translation is not only vital for translation between different languages, but also becomes inevitable when the same language is written in two or more mutually incomprehensible scripts. For example, Punjabi is written in three different scripts: Shahmukhi (a derivation of the Perso-Arabic script), Gurmukhi and Devanagari. Kazakh and Kurdish are also written in three different scripts, Arabic, Latin and Cyrillic. Malay has two writing systems, Latin and Jawi (a derivation of the Arabic script), *etc.* Figure 1 shows an example of scriptural divide between Hindi and Urdu.

دنیا کو امن کی ضرورت ہے۔

दुनिया को अमन की ज़रूरत है।

[d̪ʊnɪjɑ ko əmən ki zərurət̪ hæ.]

The world needs peace.

Figure 1: Example of scriptural divide

Thus, solving the scriptural translation problem is vital to bridge the scriptural divide between the speakers of different languages as well as of the same language.

Punjabi, Sindhi, Seraiki and Kashmiri exist on both sides of the common border between India and Pakistan and all of them are written in two or more mutually incomprehensible scripts. The Hindi–Urdu pair exists both in India and Pakistan. We call all these languages the *Indo-Pak* languages.

3 Challenges of Scriptural Translation

In this section, we describe the main challenges of scriptural translation.

3.1 Scriptural divide

There exists a written communication gap between people who can understand each other verbally but cannot read each other. They are virtually divided and become *scriptural aliens*. Examples are the Hindi & Urdu communities, the Punjabi/Shahmukhi & Punjabi/Gurmukhi communities, *etc.* An example of scriptural divide is shown in Figure 1. Such a gap also appears when people want to read some foreign language or access a bilingual dictionary and are not familiar with the writing system. For example, Japanese–French or French–Urdu dictionaries are useless for French learners because of the scriptural divide. Table 1 gives some figures on how this scriptural divide affects a large population of the world.

Sr.	Language	Number of Speakers
1	Hindi	853,000,000
2	Urdu	164,290,000
3	Punjabi	120,000,000
4	Sindhi	21,382,120
5	Seraiki	13,820,000
6	Kashmir	5,640,940
Total		**1178,133,060**

Table 1: Number of Speakers of Indo-Pak languages

3.2 Under-resourced languages

Under-resourced and under-written features of the source or target language are the second big challenge for *scriptural translation*. The lack of standard writing practices or even the absence of a standard code page for a language makes transliteration or transcription very hard. The existence of various writing styles and systems for a language leads towards a large number of variants and it becomes difficult and complex to handle them.

In the case of Indo-Pak languages, Punjabi is the largest language of Pakistan (more than 70 million) and is more a spoken language than a written one. There existed only two magazines (one weekly and one monthly) in 1992 (Rahman, 1997). In the words of (Rahman, 2004), "… *there is little development in Punjabi, Pashto, Balochi and other languages*…". (Malik, 2005) reports the first effort towards establishing a standard code page for Punjabi-Shahmukhi and till date, a standard code page for Shahmukhi does not exist. Similar problems also exist for the Kashmiri and Seraiki languages.

3.3 Absence of necessary information

There are cases where the necessary and indispensable information for scriptural translation are missing in the source text. For example, the first word دنیا [d̪ʊnɪjɑ] (world) of the example sentence of Figure 1 misses crucial diacritical information, mandatory to perform Urdu to Hindi scriptural translation. Like in Arabic, diacritical marks are part of the Urdu writing system but are sparingly used in writings (Zia, 1999; Malik *et al.*, 2008; Malik *et al.*, 2009).

Figure 2(a) shows the example word without diacritical marks and its wrong Hindi conversion according to conversion rules (explained later). The Urdu community can understand the word in its context or without the context because people are tuned to understand the Urdu text or word without diacritical marks, but the Hindi conversion of Figure 2(a) is not at all acceptable or readable in the Hindi community.

Figure 2(b) shows the example word with diacritical marks and its correct Hindi conversion according to conversion rules. Similar problems also arise for the other Indo-Pak languages. Therefore, missing information in the source text makes the scriptural translation problem computationally complex and difficult.

[ɑ] ا [j] ی [ɪ] ِ [n] ن [ʊ] ُ [d̪] د = دُ نِیا
दुनिया = द [d̪] ु [ʊ] न [n] ि [ɪ] य [j] ा[ɑ]
(b) with necessary information
[ɑ] ا [j] ی [n] ن [d̪] د = دنیا
दनया = द [d̪] न [n] य [j] ा[ɑ]
(a) without necessary information

Figure 2: Example of missing information

3.4 Different spelling conventions

Different spelling conventions exist across different scripts used for the same language or for different languages because users of a script are tuned to write certain words in a traditional way. For example, the words یہ [je] (this) = ی [j] + ہ [h] and وہ [vo] (that) = و [v] + ہ [h] are used in Urdu and Punjabi/Shahmukhi. The character ہ [h] produces the vowel sounds [e] and [o] in the example words respectively. On the other hand, the example words are written as ये [je] & वो [vo] and ਯੇ [je] & ਵੋ [vo] in Devanagari and Gurmukhi, respectively. There exist a large number of such conventions between Punjabi/Shahmukhi–Punjabi Gurmukhi, Hindi–Urdu, *etc.*

Different spelling conventions are also driven by different religious influences on different communities. In the Indian sub-continent, Hindi is a part of the Hindu identity, while Urdu is a part of the Muslim identity[1] (Rahman, 1997; Rai, 2000). Hindi derives its vocabulary from Sanskrit, while Urdu borrows its literary and scientific vocabulary from Persian and Arabic. Hindi and Urdu not only borrow from Sanskrit and Persian/Arabic, but also adopt the original spellings of the borrowed word due the sacredness of the original language. These differences make scriptural translation across scripts, dialects or languages more challenging and complex.

3.5 Transcriptional ambiguities

Character level scriptural translation across different scripts is ambiguous. For example, the Sindhi word اِنسان [ɪnsɑn] (human being) can be converted into Devanagari either as इंसान [ɪnsɑn] or इसान* [ɪsɑn] (* means wrong spellings). The transliteration process of the example word from Sindhi to Devanagari is shown in Figure 3(a). The transliteration of the third character from the left, Noon (ن) [n], is ambiguous because in the middle of a word, Noon may represent a consonant [n] or the nasalization [ɲ] of a vowel.

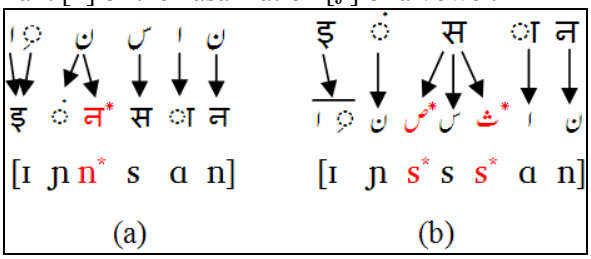

Figure 3: Sindhi transliteration example

In the reverse direction, the Sindhi Devanagari word इंसान [ɪnsɑn] can be converted into a set of possible transliterations [اِنسان, اِنصان*, اِنثان*]. All these possible transliterations have the same pronunciation [ɪnsɑn] but have different spellings in

[1] The Hindi movement of the late 19th century played a central role in the ideologization of Hindi. The movement started in reaction to the British Act 29 of 1837 by which Persian was replaced by Hindustani/Urdu, written in Persian script, as the official vernacular of the courts of law in North India. It is the moment in history, when Hindi and Urdu started to emerge as Hindu and Muslim identities.

the Perso-Arabic script, as shown in Figure 3(b). Similar kinds of ambiguities also arise for other pairs of scripts, dialects or languages. Thus these ambiguities increase the complexity and difficulty of *scriptural translation*.

3.6 Distinctive sound inventories

Sound inventories across dialects or languages can be different. Consider the English–Japanese pair. Japanese make no distinction between the 'L' [l] and 'R' [r] sounds so that these two English sounds collapse onto the same Japanese sound (Knight and Al-Onaizan, 1998).

For Indo-Pak languages, Punjabi/Gurmukhi (a dialect of Punjabi spoken in India) possesses two additional sounds than Punjabi/Shahmukhi (a dialect of Punjabi spoken in Pakistan). Similarly, Hindi, Punjabi, Sindhi and Seraiki have the retroflex form [ɳ], but Urdu and Kashmiri do not. Marathi has 14 vowels in contrast to Hindi's 11 vowels, shown in Table 2.

Hindi Vowels
अ [ə] आ [ɑ] इ [ɪ] ई [i] उ [ʊ] ऊ [u] ऋ [r̥] ए [e] ऐ [æ] ओ [o] औ [ɔ]
Marathi Vowels
अ [ə] आ [ɑ] इ [ɪ] ई [i] उ [ʊ] ऊ [u] ऋ [r̥] ए [e] ऐ [æ] ओ [o] औ [ɔ] अं [əŋ] अः [əh] ऌ [l̥]

Table 2: Hindi and Marathi vowel comparison

Scriptural translation approximates the pronunciation of the source language or dialect in the target due to different sound inventories. Thus a distinctive sound inventory across scripts, dialects or languages increases ambiguities and adds to the complexity of the *scriptural translation* problem.

4 Universal Intermediate Transcription

UIT (Universal Intermediate Transcription) is a multipurpose pivot. In the current study, it is used as a *phonetico-morphotactic* pivot for the *surface morphotactic translation* or scriptural translation.

Although we have not used IPA as encoding scheme, we have used the IPA coding associated with each character as the encoding principle for our ASCII encoding scheme. We selected the printable ASCII characters to base the UIT encoding scheme because it is universally portable to all computer systems and operating systems without any problem (Boitet and Tchéou, 1990; Hieronymus, 1993; Wells, 1995). UIT is a deterministic and unambiguous scheme of transcription for Indo-Pak languages in ASCII range 32–126, since a text in this rage is portable across computers and operating systems (Hieronymus, 1993; Wells, 1995).

Speech Assessment Methods Phonetic Alphabet (SAMPA)[2] is a widely accepted scheme for encoding IPA into ASCII. The purpose of SAMPA was to form the basis of an international standard machine-readable phonetic alphabet for the purpose of international collaboration in speech research (Wells, 1995). The UIT encoding of Indo-Pak languages is developed as an extension of the SAMPA and X-SAMPA that covers all symbols on the IPA chart (Wells, 1995).

4.1 UIT encodings

All characters of the Indo-Pak languages are subdivided into three categories, consonants, vowels and other symbols (punctuations and digits).

Consonants are further divided into aspirated consonants and non-aspirated consonants. For aspiration, in phonetic transcription a simple 'h' following the base consonant symbol is considered adequate (Wells, 1995). In the Indo-Pak languages, we have two characters with IPA [h]. Thus to distinguish between the 'h' consonants and the aspiration, we use *underscore* '_' to mark the aspirate and we encode an aspiration as '_h'. For example, the aspirated consonants ٹھ [t^h], پھ [p^h] and ڇھ [$ʧ^h$] of the Indo-Pak languages are encoded as 't`_h', 'p_h' and 't_S_h' respectively. Similarly for the dental consonants, we use the '_d' marker. For example, the characters د [d̪] and ت [t̪] are encoded as 'd_d' and 't_d' in UIT. Table 3 shows the UIT encodings of Hindi and Urdu aspirated consonants.

Hindi	Urdu	UIT	Hindi	Urdu	UIT
भ	بھ [b^h]	b_h	ह्र	رھ [r^h]	r_h
फ	پھ [p^h]	p_h	ढ़	ڑھ [$ɽ^h$]	r`_h
थ	تھ [$t̪^h$]	t_d_h	ख	کھ [k^h]	k_h
ठ	ٹھ [$ʈ^h$]	t`_h	घ	گھ [g^h]	g_h
झ	جھ [$ʤ^h$]	d_Z_h	ल्ह	لھ [l^h]	l_h
छ	چھ [$ʧ^h$]	t_S_h	म्ह	مھ [m^h]	m_h

[2] http://www.phon.ucl.ac.uk/home/sampa/

ध	دھ [d̪ʰ]	d_d_h	न्ह	نھ [nʰ]	n_h
ढ	ڈھ [ɖʰ]	d`_h			

Table 3: UIT encodings of Urdu aspirated consonants

Similarly, we can encode all characters of Indo-Pak languages. Table 4 gives UIT encodings of Hindi and Urdu non-aspirated consonants. We cannot give all encoding tables here due to shortage of space.

Hindi	Urdu	UIT	Hindi	Urdu	UIT
ब	ب [b]	b	स	ص [s]	s2
प	پ [p]	p	ज़	ض [z]	z2
त	ت [t̪]	t_d	त	ط [t̪]	t_d1
ट	ٹ [ʈ]	t`	ज़	ظ [z]	z3
स	ٹ [s]	s1	-	ع [ʔ]	?
ज	ج [ʤ]	d_Z	ग़	غ [ɣ]	X
च	چ [ʧ]	t_S	फ़	ف [f]	f
ह	ح [h]	h1	क़	ق [q]	q
ख़	خ [x]	x	क	ک [k]	k
द	د [d̪]	d_d	ग	گ [g]	g
ड	ڈ [ɖ]	d`	ल	ل [l]	l
ज़	ذ [z]	z1	म	م [m]	m
र	ر [r]	r	न	ن [n]	n
ड़	ڑ [ɽ]	r`	व	و [v]	v
ज़	ز [z]	z	ह	ہ [h]	h
ज़	ژ [ʒ]	Z	य	ی [j]	j
स	س [s]	s	त	ۃ [t̪]	t_d2
श	ش [ʃ]	S	ण	- [ɳ]	n`
ष	ش [ʃ]	S1	ं	ں [ŋ]	~

Table 4: UIT encodings of Urdu non-aspirated consonants

5 Finite-state Scriptural Translation Model

Figure 4 shows the system architecture of our finite-state scriptural translation system.

`Text Tokenizer` receives and converts the input source language text into constituent words or tokens. This list of the source language tokens is then passed to the `UIT Encoder` that encodes these tokens into a list of UIT tokens using the source language to UIT conversion transducer from the repertoire of *Finite-State Transducers*. These UIT tokens are given to the `UIT Decoder` that decodes them into target language tokens using the UIT to target language conversion transducer from the repertoire of Transducers. Finally, `Text Generator` generates the target language text from the translated target language tokens.

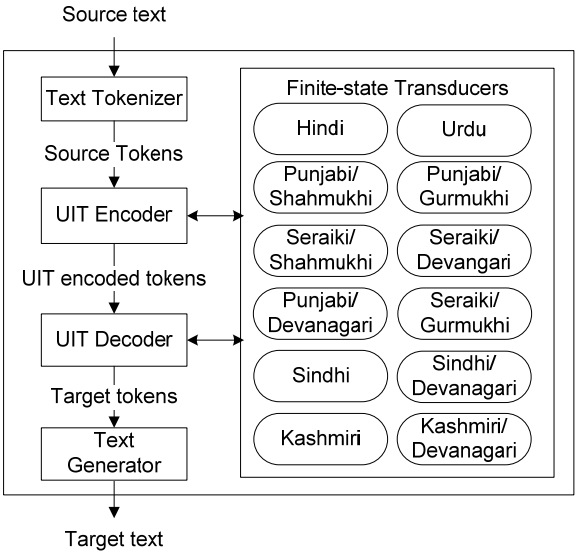

Figure 4: System Architecture of fintie-state scriptural translation

5.1 Finite-state Transducers

Both conversions of the source language text into the UIT encoded text and from the UIT encoded text into the target language text are regular relations on strings. Moreover, regular relations are closed under serial composition and a finite set of conversion relations when applied to each other's output in a specific order, also defines a regular expression (Kaplan and Kay, 1994). Thus we model the conversions from the source language to UIT and from UIT to the target language as finite-state transducers. These translational transducers can be deterministic and non-deterministic.

Character Mappings: Table 5 shows regular relations for converting Hindi and Urdu aspirated consonants into UIT.

IPA	Hindi to UIT	Urdu to UIT
b^h	भ → b_h	ﮪ → b_h
p^h	फ → p_h	ﭫ → p_h
$\underset{\frown}{t}^h$	थ → t_d_h	ﺗ → t_d_h
$ʈ^h$	ठ → t`_h	ﭨ → t`_h
$ʤ^h$	झ → d_Z_h	ﺟ → d_Z_h
$ʧ^h$	छ → t_S_h	ﭽ → t_S_h

$d̪^h$	ध → d_d_h	دھ → d_d_h
$ɖ^h$	ढ → d`_h	ڈھ → d`_h
r^h	ह़ → r_h	رھ → r_h
$ɽ^h$	ढ़ → r`_h	ڑھ → r`_h
k^h	ख → k_h	کھ → k_h
g^h	घ → g_h	گھ → g_h
l^h	ल्ह → l_h	لھ → l_h
m^h	म्ह → m_h	مھ → m_h
n^h	न्ह → n_h	نھ → n_h

Table 5: Regular rules for aspirated consonants of Hindi and Urdu

By interchanging the UIT encodings before the arrow sign and the respective characters of Hindi and Urdu after the arrow, we can construct regular conversion relations from UIT to Hindi and Urdu. We have used XFST (Xerox finite-state engine) to build finite-state transducers. Table 6 shows a sample XFST code.

Contextual Mappings: A contextual mapping is a contextual rule that determines a desired output when a character appears in a certain context. The third command of Table 6 models another contextual mapping saying that 'ह्' is translated by '_h' when it is preceded by any of the characters र, ल, म, and न. The second last rule of Table 6 models the contextual mapping rule that 'A1' is translated into 'ہ' when it is at the end of a word and preceded by a consonant.

```
clear stack
set char-encoding UTF-8
read regex [ि -> I];
read regex [ख -> [k "_" h], घ -> [g
  "_" h], छ -> [t "_" S "_" h], झ -
  > [d "_" Z "_" h], ठ -> [t "`" "_"
  h], ढ -> [d "`" "_" h], थ -> [t
  "_" d "_" h], ध -> [d "_" d "_"
  h], फ -> [p "_" h], भ -> [b "_"
  h], ढ़ -> [r "`" "_" h], स -> s, त
  -> [t "_" d], र -> r, ल -> l, म ->
  m, न -> n, व -> v, ह -> h];
read regex [[् ह] -> ["_" h] || [र |
ल | म | न] _ ];
compose net
```

Table 6: Sample XFST code

Vowel representations in Urdu, Punjabi/Shahmukhi, Sindhi, Seraiki/Shahmukhi and Kashmiri are highly context-sensitive (Malik *et al.*, 2010).

6 Experiments and Results

A sample run of our finite-state scriptural translation system on the Hindi to Urdu example sentence of Figure 1 is shown in Table 7.

Text Tokenizer	UIT Encoder	UIT Decoder	
		Unique output	*Ambiguous outputs*
दुनिया	dUnIjA1	دُنِیا	[دُنِیہ , دُنیا]
को	ko	کو	[قو , کو]
अमन	@mn	امن	[امن]
की	ki	کی	[قی , کی]
ज़रूरत	zrurt_d	زرُوت	[زرُرت , ضرُورت , ذرُورت , ظرُورت , ژرُورت , …]
है	h{	ہَے	[حَے , ہے]

Table 7: Sample run of finite-state scriptural translation model on Hindi to Urdu example

`Text Generator` converts the unique output of the UIT Decoder into an Urdu sentence with one error in the fifth word (highlighted), shown in Figure 5.

دنِیا کو امن کی زرورت ہے

Figure 5: Unique output of the sample run by deterministic FSTs

On the other hand, from the ambiguous output of the UIT Decoder, we can generate 240 output sentences, but only one is the correct scriptural translation of the source Hindi sentence in Urdu. The correct sentence is shown in Figure 6. The sole difference between the output of the deterministic FST and the correct scriptural translation is highlighted in both sentences shown in Figure 5 and 6.

دنِیا کو امن کی ضرورت ہے

Figure 6: Correct scriptural translation of the example

6.1 Test Data

Table 8 shows test sets for the evaluation of our finite-state scriptural translation system.

Data set	Language pair	No. of words	No. of sentences	Source
HU 1	Hindi–Urdu	52,753	-	Platts dictionary
HU 2	Hindi–Urdu	4,281	200	Hindi corpus
HU 3	Hindi–Urdu	4,632	226	Urdu corpus
PU	Punjabi/Shahmukhi–Punjabi/Gurmukhi	5,069	500	Classical poetry
SE	Seraiki/Shahmukhi–Seraiki/Devanagari	2,087	509	Seraiki poetry

Table 8: Test Sets of Hindi, Urdu, Punjabi and Seraiki

HU 1 is a word list obtained from the Platts dictionary[3] (Platts, 1884).

6.2 Results

For Hindi to Urdu scriptural translation, we have applied the finite-state model to all Hindi inputs of HU Test sets 1, 2 and 3. In general, it gives us an Urdu output with the necessary diacritical marks. To evaluate the performance of Hindi to Urdu scriptural translation of our finite-state system against the Urdu without diacritics, we have created a second Urdu output by removing all diacritical marks from the default Urdu output of the finite-state system. We have calculated the *Word Accuracy Rate* (WAR) and *Sentence Accuracy Rate* (SAR) for the default and the processed Urdu outputs by comparing them with the Urdu references with and without diacritics respectively. To compute WAR and SAR, we have used the SCLITE utility from the Speech Recognition Scoring Toolkit (SCTK)[4] of NIST. The results of Hindi to Urdu scriptural translation are given in Table 24.

Test Set	Default output		Processed output	
	Word Level	Sentence Level	Word Level	Sentence Level
HU 1	32.5%	-	78.9%	-
HU 2	90.8%	26.5%	91.0%	27%
HU 3	81.2%	8.8%	82.8%	9.7%

Table 9:Hindi to Urdu scriptural translation restuls

The finite-state scriptural translation system for Hindi to Urdu produces an Urdu output with diacritics. However, we know that the Urdu community is used to see the Urdu text without diacritics. Thus, we removed all diacritical marks from the Urdu output text that is more acceptable to the Urdu community. By this post-processing, we gain more than 40% accuracy in case of HU Test Set 1. We also gain in accuracy for the other test sets.

For the classification of our scriptural translation systems, we have devised two scales. One corresponds to the word accuracy rate and the other corresponds to the sentence level accuracy. They are shown in Figure 7 and 8.

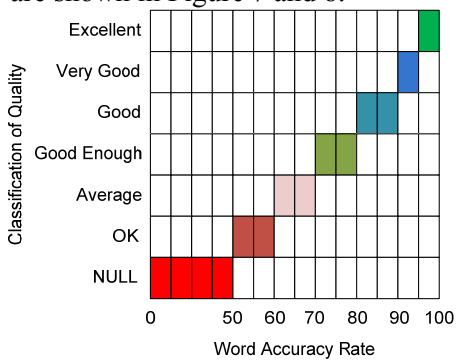

Figure 7: Classification scale based on the word accuracy rate for scriptural transaltion

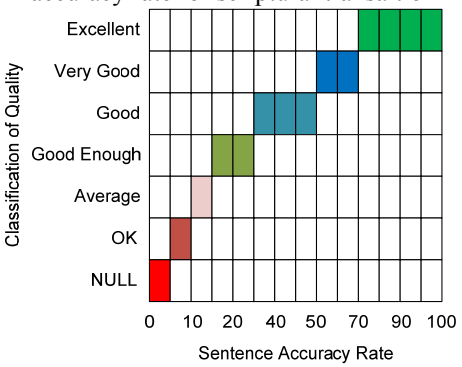

Figure 8: Classification scale based on the sentence accucary rate for scriptural translation

According to the scale of Figure 7 and 8, the Hindi to Urdu scriptural translation system is classified as 'Good' and 'Good Enough', respectively.

The subjective evaluations like usability, effectiveness and adequacy depend on several factors. A user with a good knowledge of Hindi and Urdu languages would rate our Hindi to Urdu system quite high and would also rate the Urdu output very usable. Another user who wants to read a Hindi text, but does not know Hindi, would also rate this system and the Urdu output quite high and very usable respectively, because it serves its purpose.

On the other hand, a user who wants to publish a Hindi book in Urdu, would rate this system not very good. This is because he has to localize the Hindi vocabulary of Sanskrit origin as the acceptance of the Hindi vocabulary in the Urdu

[3] Shared by University of Chicago for research purposes.

[4] http://www.itl.nist.gov/iad/mig//tools/

community, target of his published book, is very low. Thus the subjective evaluation depends on various factors and it is not easy to compute such measures for the evaluation of a scriptural translation system, but they are vital in real life.

For Urdu to Hindi scriptural translation, we have two inputs for each HU Test Set. One input contains all diacritical marks and the other does not contain any. On Hindi side, we have a single Hindi reference with which we will compare both Hindi outputs. We already know that it will give us less accuracy rates for the Urdu input without diacritical marks that are mandatory for correct Urdu to Hindi scriptural translation. The results for Urdu to Hindi scriptural translation are given in Table 10.

Test Set	With diacritics		Without diacritics	
	Word Level	Sentence Level	Word Level	Sentence Level
HU 1	68.0%	-	31.2%	-
HU 2	83.9%	10%	53.0%	1%
HU 3	98.4%	73.9%	58.9%	0.4%

Table 10: Urdu to Hindi scriptural translation results

For the Urdu input with diacritics, the accuracy of the Urdu to Hindi finite-state scriptural translation system is 83.9% at word level for HU Test Set 2 and it is classified as 'GOOD' the classification scale of Figure 7. On the other hand, it shows a sentence-level accuracy of 10% for the same test set and is classified as 'AVERAGE' by the classification scale of Figure 8.

For the Urdu input without diacritics, the Urdu to Hindi scriptural translation system is classified as 'OK' by the scale of Figure 7 for HU Test set 2 and 3. It is classifies as 'NULL' for HU Test Set 1. According to the scale of Figure 8, it is classified as 'NULL' for all three test sets.

For Punjabi scriptural translation, we also developed two types of output default and processed for Gurmukhi to Shahmukhi translation. In the reverse direction, it has two types of inputs, one with diacritics and the other without diacritics. Table 11 and 12 shows results of Punjabi scriptural translation.

Test Set	Default output		Processed output	
	Word Level	Sentence Level	Word Level	Sentence Level
PU	84.2%	27.8%	85.2%	29.9%

Table 11: Gurmukhi to Shahmukhi scriptural translation results

Test Set	With diacritics		Without diacritics	
	Word Level	Sentence Level	Word Level	Sentence Level
PU	98.8%	90.3%	67.3%	6.4%

Table 12: Shahmukhi to Gurmukhi scriptural translation results

Compared to the Hindi–Urdu pair, the Punjabi/Shahmukhi–Punjabi/Gurmukhi pair is computationally less hard. The post-processing to the default out of the finite-state scriptural translation systems for Punjabi/Gurmukhi to Punjabi/Shahmukhi also helps to gain an increase of approximately 1% and 2% at word and sentence levels respectively. The Shahmukhi to Gurmukhi scriptural translation system is classified as 'GOOD' by both scales of Figure 7 and 8. Thus the usability of the Punjabi finite-state scriptural translation system is higher than the Hindi–Urdu finite-state scriptural translation system.

In the reverse direction, the Shahmukhi to Gurmukhi scriptural translation system gives an accuracy of 98.8% and 67.3% for the Shahmukhi input text with and without diacritics respectively. For the Shahmukhi input text with diacritics, the scriptural translation system is classified as 'EXCELLENT' by both scales. On the other hand, it is classified as 'NULL' according to the scale of Figure 8 for the Shahmukhi input text without diacritical marks.

Similar to Hindi–Urdu and Punjabi finite-state scriptural translation, we have applied our finite-state system to the Seraiki test set. Here again, we have developed a processed Seraiki/Shahmukhi output from the default output of our finite-state system by removing the diacritics. The results are given in Table 13 and 14.

Test Set	Default output		Processed output	
	Word Level	Sentence Level	Word Level	Sentence Level
SE	81.3%	19.4%	83.7%	20.3%

Table 13: Seraiki/Devanagari to Seraiki/Shahmukhi scriptural translation results

Test Set	With diacritics		Without diacritics	
	Word Level	Sentence Level	Word Level	Sentence Level
SE	95.2%	76.4%	58.6%	8.6%

Table 14: Seraiki/Shahmukhi to Seraiki/Devanagari scriptural translation results

In the case of the Seraiki/Devanagari to Seraiki/Shahmukhi scriptural translation system, the post-processing also helps to gain an increase in word accuracy of approximately 1 to 2 percent

both at the word and the sentence levels. The accuracy for both the default and the processed Seraiki/Shahmukhi outputs is also more than 80% at word level. The system is classified as 'GOOD' and 'GOOD ENOUGH' according to the scale of Figure 7 and 8 respectively.

The absence of diacritical marks in the Seraiki/Shahmukhi has a very bad effect on the accuracy of the finite-state scriptural translation system. The scriptural translation system is classified as 'NULL' for the Seraiki/Shahmukhi input text without diacritics.

7 Conclusion

Finite-state methods are robust and efficient to implement scriptural translation rules in a very precise and compact manner.

The missing information and the diacritical marks in the source text proved to be very critical, crucial and important for achieving high and accurate results. The above results support our hypothesis that lack of important information in the source texts considerably lowers the quality of scriptural translation. They are crucial and their absence in the input texts decreases the performance considerably, from more than 80% to less than 60% at word level. Thus restoration of the missing information and the diacritical marks or reducing the effect of their absence on the scriptural translation is one of the major questions for further study and work.

In general, only word accuracy rates are reported. We have observed that only word accuracy rates may depict a good performance, but the performance of the same system at sentence-level may be not very good. Therefore, subjective evaluations and usage of translation results in real life should also be considered while evaluating the translation quality.

Acknowledgments

This study is supported by Higher Education Commission (HEC), Government of Pakistan under its overseas PhD scholarship scheme. We are also thankful to Digital South Asian Library, University of Chicago for sharing Platts dictionary data (Platts, 1884).

References

AbdulJaleel, N. and L. S. Larkey. 2003. Statistical Transliteration for English-Arabic Cross Language Information Retrieval. 12th international Conference on information and Knowledge Management (CIKM 03), New Orleans. 139-146.

Al-Onaizan, Y. and K. Knight. 2002. Machine Transliteration of Names in Arabic Text. Workshop on Computational Approaches To Semitic Languages, the 40th Annual Meeting of the ACL, Philadelphia, Pennsylvania, 1-13.

Arbabi, M., S. M. Fischthal, V. C. Cheng and E. Bart. 1994. Algorithms for Arabic Name Transliteration. *IBM J. Res. Dev.* 38(2): 183-193.

Boitet, C. and F. X. Tchéou. 1990. On a Phonetic and Structural Encoding of Chinese Characters in Chinese texts. ROCLING III, Taipeh. 73-80.

Fujii, A. and T. Ishikawa. 2001. Japanese/English Cross-Language Information Retrieval: exploration of query translation and transliteration. *Computers and the Humanities* 35(4): 389-420.

Halpern, J. 2002. Lexicon-based Orthographic Disambiguation in CJK Intelligent Information Retrieval. 3rd workshop on Asian language resources and international standardization, the 19th International Conference on Computational Linguistics (COLING), Taipei, Taiwan. 1-7.

Hieronymus, J. 1993. ASCII Phonetic Symbols for the World's Languages: Worldbet. AT&T Bell Laboratories.

Jeong, K. S., S. H. Myaeng, J. S. Lee and K.-S. Choi. 1999. Automatic Identification and Back-transliteration of Foreign Words for Information Retrieval. *Information Processing and Management* 35: 523-540.

Kang, B. and K. Choi. 2000. Automatic Transliteration and Back Transliteration by Decision Tree Learning. 2nd International Conference on Evaluation and Language Resources (ELRC), Athens.

Kaplan, R. M. and M. Kay. 1994. Regular Models of Phonological Rule Systems. 20(3).

Knight, K. and Y. Al-Onaizan. 1998. Translation with Finite-State Devices 3rd Conference of the Association for Machine Translation in the Americas on Machine Translation and the Information Soup (AMTA-98), Pennsylvania. 421-437.

Lee, J. S. and K. S. Choi. 1998. English to Korean Statistical Transliteration for Information Retrieval. *Computer Processing of Oriental languages* 12(1): 17-37.

Malik, M. G. A. 2005. Towards a Unicode Compatible Punjabi Character Set. 27th Internationalization and Unicode Conference, Berlin.

Malik, M. G. A., L. Besacier, C. Boitet and P. Bhattacharyya. 2009. A Hybrid Model for Urdu Hindi Transliteration. Joint conference of the 47th Annual Meeting of the Association of Computational Linguistics and the 4th

International Joint Conference on Natural Language Processing of the Asian Federation of NLP ACL/IJCNLP Workshop on Named Entities (NEWS-09), Singapore, 177–185.

Malik, M. G. A., C. Boitet and P. Bhattacharyya. 2008. Hindi Urdu Machine Transliteration using Finite-state Transducers. 22nd International Conference on Computational Linguistics (COLING), Manchester, 537-544.

Malik, M. G. A., C. Boitet and P. Bhattacharyya. 2010. Analysis of Noori Nast'aleeq for Major Pakistani Languages. 2nd Workshop on Spoken Language Technologies for Under-resourced Languages SLTU-2010, Penang, Malaysia.

Mohri, M. 1997. Finite-state Transducers in Language and Speech Processing. 23(2).

Pirkola, A., J. Toivonen, H. Keskustalo, K. Visala and K. Järvelin. 2003. Fuzzy Translation of Cross-lingual Spelling Variants. 26th Annual international ACM SIGIR Conference on Research and Development in informaion Retrieval, Toronto.

Platts, J. T. 1884. A Dictionary of Urdu, Classical Hindi and Englsih. W. H. Allen & Co.

Rahman, T. 1997. *Language and Politics in Pakistan*. Oxford University Press, Lahore.

Rahman, T. 2004. Language Policy and Localization in Pakistan: Proposal for a Paradigmatic Shift. Crossing the Digital Divide, SCALLA Conference on Computational Linguistics, Katmandu.

Rai, A. 2000. *Hindi Nationalism*. Orient Longman Private Limited, New Delhi.

Roche, E. and Y. Schabes, Eds. 1997. Finite-state Language Processing. MIT Press, Cambridge.

Sakai, T., A. Kumano and T. Manabe. 2002. Generating Transliteration Rules for Cross-language Information Retrieval from Machine Translation Dictionaries. IEEE Conference on Systems, Man and Cybernatics.

Stall, B. and K. Knight. 1998. Translating Names and Technical Terms in Arabic Text. Workshop on Computational Approaches to Semitic Languages, COLING/ACL, Montreal, 34-41.

Virga, P. and S. Khudanpur. 2003. Transliteration of Proper Names in Cross-language Applications. 26th Annual international ACM SIGIR Conference on Research and Development in informaion Retrieval, Toronto.

Wells, J. C. 1995. Computer-coding the IPA: a proposed extension of SAMPA. University College London.

Yan, Q., G. Gregory and A. E. David. 2003. Automatic Transliteration for Japanese-to-English Text Retrieval. 26th annual international ACM SIGIR conference on Research and development in information retrieval, 353-360.

Zia, K. 1999. Standard Code Table for Urdu. 4th Symposium on Multilingual Information Processing (MLIT-4), Yangon.