A Hybrid Model for Urdu Hindi Transliteration

Abbas Malik Laurent Besacier Christian Boitet
GETALP, Laboratoire d'Informatique Grenoble (LIG)
Université Joseph Fourier
Abbas.Malik, Laurent.Besacier,
Christian.Boitet@imag.fr

Pushpak Bhattacharyya
IIT Bombay
pb@cse.iitb.ac.in

Abstract

We report in this paper a novel hybrid approach for Urdu to Hindi transliteration that combines *finite-state machine (FSM)* based techniques with *statistical word language model* based approach. The output from the FSM is filtered with the word language model to produce the correct Hindi output. The main problem handled is the case of omission of diacritical marks from the input Urdu text. Our system produces the correct Hindi output even when the crucial information in the form of diacritic marks is absent. The approach improves the accuracy of the transducer-only approach from 50.7% to 79.1%. The results reported show that performance can be improved using a word language model to disambiguate the output produced by the transducer-only approach, especially when diacritic marks are not present in the Urdu input.

1 Introduction

Transliteration is a process to transcribe a word written in one language, in another language by preserving its articulation. It is crucial for handling out-of-vocabulary (OOV) words in different domains of Natural Language Processing (NLP), especially in Machine Translation (Knight and Graehl, 1998; Knight and Stall, 1998; Paola and Sanjeev, 2003), Cross-Lingual Information Retrieval (Pirkola *et al.*, 2003), the development of multi-lingual resources (Yan *et al.*, 2003) and multi-lingual text and speech processing. It is also useful for Inter-dialectal translation without lexical changes and sometimes it is mandatory when the dialects in question use mutually incomprehensible writing systems. Such cases exists in Malay (written in 2 different scripts), Turkish (2 scripts), Kurdish (3 scripts), Hindi/Urdu (2 scripts), Punjabi (2 scripts), *etc*., where words are transliterated from one script to the other, irrespective of their type (noun, verb, *etc*., and not only proper nouns and unknown words). In this study, we will focus on Hindi/Urdu example.

Hindi and Urdu are written in two mutually incomprehensible scripts, Devanagari and Urdu script – a derivative of Persio-Arabic script respectively. Hindi and Urdu are the official languages of India and the later is also the National language of Pakistan (Rahman, 2004). Table 1 gives an idea about the number of speakers of Hindi and Urdu.

	Native Speaker	2nd Lang. Speaker	Total
Hindi	366	487	853
Urdu	60.29	104	164.29
Total	426.29	591	1,017.29
Source: (Grimes, 2000) all numbers are in millions			

Table 1: Hindi and Urdu Speakers

Notwithstanding the transcriptional differences, Hindi and Urdu share phonology, grammar, morphology, literature, cultural heritage, *etc*. People from Hindi and Urdu communities can understand the verbal expressions of each other but the written expression of one community is alien to the other community.

A finite-state transliteration model for Hindi and Urdu transliteration using the Universal Intermediate Transcription (UIT – a pivot between the two scripts) was proposed by Malik *et al.* (2008). The non-probabilistic finite-state model is not powerful enough to solve all problems of Hindi ↔ Urdu transliteration. We visit and analyze Hindi ↔ Urdu transliteration problems in the next section and show that the solution of these problems is beyond the scope of a non-probabilistic finite-state transliteration model.

Proceedings of the 2009 Named Entities Workshop, ACL-IJCNLP 2009, pages 177–185,
Suntec, Singapore, 7 August 2009.

Following this, we show how a statistical model can be used to solve some of these problems, thereby enhancing the capabilities of the finite-state model.

Thus, we propose a hybrid transliteration model by combining the finite-state model and the *statistical word language model* for solving Hindi ↔ Urdu transliteration problems, discussed in section 2. Section 3 will throw light on the proposed model, its different components and various steps involved in its construction. In section 4, we will report and various aspects of different experiments and their results. Finally, we will conclude this study in section 5.

2 Hindi Urdu Transliteration

In this section, we will analyze Hindi ↔ Urdu transliteration problems and will concentrate on Urdu to Hindi transliteration only due to shortage of space and will discuss the reverse transliteration later. Thus, the remainder of the section analyzes the problems from Urdu to Hindi transliteration.

2.1 Vowel, Yeh (ی) and Waw (و)

Urdu is written in a derivation of Persio-Arabic script. Urdu vowels are represented with the help of four long vowels Alef-madda (آ), Alef (ا), Waw (و), Yeh (ی) and diacritical marks. One vowel can be represented in many ways depending upon its context or on the origin of the word, e.g. the vowel [ɑ] is represented by Alef-madda (آ) at the beginning of a word, by Alef (ا) in the middle of a word and in some Persio-Arabic loan word, it is represented by the diacritical mark Khari Zabar (ٰ). Thus Urdu has very complex vowel system, for more details see Malik *et al.* (2008). Urdu contains 10 vowels, and 7 of them also have their nasalization forms (Hussain, 2004; Khan, 1997) and 15 diacritical marks. Thou diacritical marks form the cornerstone of the Urdu vowel system, but are sparingly used (Zia, 1999). They are vital for the correct Urdu to Hindi transliteration using the finite-state transliteration model. The accuracy of the finite-state transliteration model decreases from above 80% to 50% in the absence of diacritical marks. Figure 1 shows two example Urdu phrases (i) with and (ii) without the diacritical marks and their Hindi transliteration using the finite-state transliteration model. Due to the absence of Zabar (ٰ) in the first and the last words in (1)(ii) and in the 5th word in (2)(ii), vowels ै [æ] and औ [ɔ] are transliterated into vowels े [e] and ओ [o] respectively. Similarly, due to the absence of Pesh (ُ) and Zer (ِ) in 3rd and 4th words respectively in (1)(ii), both vowels ु [ʊ] and ि [ɪ] are converted into the vowel [ə]. All wrongly converted words are underlined.

(1)
(i) مَیں نے بہُت ادھِک کام نہیں کِیا ہَے
(ii) میں نے بہت ادھک کام نہیں کیا ہے
(i) मैं ने बहुत अधिक काम नहीं किया है
(ii) <u>में</u> ने <u>बहत</u> <u>अधक</u> काम <u>नहें</u> <u>कया</u> <u>हे</u>
I have not done a lot of work

(2)
(i) کیِنْدْرِیّہ سْطر پر بھی اَور راجْیہ سْطر پر بھی
(ii) کیندریہ سطر پر بھی اور راجیہ سطر پر بھی
(i) केन्द्रीय स्तर पर भी और राज्य स्तर पर भी
(ii) <u>केंदरय</u> <u>सतर</u> पर भी <u>ओर</u> <u>राजय</u> <u>सतर</u> पर भी
Both at the central level and at the state level

Figure 1: Example Urdu Phrases

In Hindi, each vowel is represented by a character and a vowel sign except the vowel [ə], which is only represented by the character अ and do not have a vowel sign (Malik *et al.*, 2008). Table 2 gives all vowel conversion problems.

Sr.	IPA	Vowel Conversion Problems	Hindi
1	ɪ	ɪ → ə	इ or ि → अ or 0*
2	ʊ	ʊ → ə	उ or ु → अ or 0*
3	i	i → e	ई or ी → ए or े
4	æ	æ → e	ऐ or ै → ए or े
5	u	u → o	ऊ or ू → ओ or ो
6	ɔ	ɔ → o	औ or ौ → ओ or ो
7	j	j → e	य → े
8	v	v → o	व → ो
* Zero (0) means deleted.			

Table 2: Vowel Problems from Urdu to Hindi

Long vowels Yeh (ی) [j] and Waw (و) [v] are also used as consonants and certain contextual rules help us to decide whether they are used as a consonant or as a vowel, *e.g.*, Yeh (ی) and Waw (و) are used as consonants at the start of a word and after the long vowel Alef-madda (آ), *etc*. Fi-

nite-state transliteration model can exploit such contextual rules but it is not possible to decide Yeh (ی) and Waw (و) as consonants in the absence of diacritics. Thus a finite-state transliteration model wrongly converts consonant Yeh (ی) and Waw (و) into vowels ے [e] and ो [o], also given in Table 2, instead of consonants Ya (य) and Wa (व) respectively, *e.g.*, in the word کنور (prince) [kʊɲvr], Waw is wrongly converted into the vowel [o] due to the absence of Zabar (ً) after it and the word becomes [kʊnor], which is not a valid word of Hindi/Urdu.

2.2 Native Sounds

The Hindi writing system contains some native sounds/characters, *e.g.*, vocalic R (ऋ) [ɾ̩], retroflex form of Na (ण) [ɳ], *etc*. On the other hand Urdu does not have their equivalents. Thus words containing such sounds are transcribed in Urdu with their approximate phonetic equivalents. All such cases are problematic for Urdu to Hindi transliteration and are given in Table 3.

Sr.	IPA	Hindi	Urdu
1	ɾ̩	ऋ or ृ	ر [r]
2	ɳ	ण	ن [n]
3	ʃ	ष	ش [ʃ]
4	Half h	ः	ہ [h]

Table 3: Sounds of Sanskrit Origin

2.3 Conjunct Form

The Hindi alphabet is partly syllabic because each consonant inherits the vowel [ə]. Two or more consonants may be combined together to form a cluster called Conjunct that marks the absence of the inherited vowel [ə] between consonants (Kellogg, 1872; Montaut, 2004). Conjunction is also used to represent the gemination of a consonant, *e.g.,* क[k]+्+क[k]=क्क[kk] where ् is the conjunct marker and aspiration of some consonants like न [n], म [m], र [r] and ल [l] when used as conjunction with ह [h], *e.g.*, न[n] + ् + ह[h] = न्ह[n^h]. Conjunction has a special meaning but native speakers use conjunct forms without any explicit rule (Montaut, 2004).

On the other hand, Urdu uses Jazam (ْ – a diacritic) and Shadda (ّ) to mark the absence of a vowel between two consonants and gemination of a consonant respectively. In the absence of these diacritics in the input Urdu text, it is not possible to decide on the conjunct form of consonants except in the case of aspiration. In Urdu, aspiration of a consonant is marked with the special character Heh-Doachashmee (ھ) (Malik *et al*., 2008), thus a finite-state transducer can easily decide about the conjunction for aspiration with a simple contextual rule, e.g. the word دُلھن (bride) [ɖʊlʰn] is correctly transliterated by our finite-state transliteration model into दुल्हन.

2.4 Native Hindi Spellings and Sanskritized Vocabulary

Sanskrit highly influences Hindi and especially its vocabulary. In some words of Sanskrit origin, the vowel ी [i] and ू [u] are transcribed as ि [ɪ] and ु [ʊ] respectively at the end of a word. Javaid and Ahmed (2009) have pointed to this issue in these words "Hindi language can have words that end on short vowel…". Table 4 gives some examples of such native words. On the other hand in Urdu, short vowels can never come at the end of a word (Javaid and Ahmed, 2009; Malik *et al*., 2008).

Vowel	Examples
ी [i]	व्यक्ति – ویکتی (person) [vjəkti] संस्कृति – سنسکرتی (culture) [səɲskrɾ̩ti] उच्चकोटि – اُچکوٹی (high) [ʊʧʧkoʈi]
ू [u]	हेतु – ہیتُو (for) [heʈu] किन्तु – کنتُو (but) [kɪnʈu] धातु – دھاتُو (metal) [ɖʱaʈu]

Table 4: Hindi Word with Short vowel at End

It is clear from above examples that short vowels at the end of a Hindi word can easily be transliterated in Urdu using a contextual rule of a finite-state transducer, but it is not possible to do so for Urdu to Hindi transliteration using a non-probabilistic finite-state transliteration model. Thus Urdu to Hindi transliteration can also be

considered as a special case of Back Transliteration.

In some words, the vowel ◌ُ [u] is written as the vowel ◌ु [ʊ], *e.g.*, हुए – ہُوئے or हुआ – ہُوا (to be) [hue], राजनपुर (name of a city) [radʒənpur]. Some of these cases are regular and can be implemented as contextual rules in a finite-state transducer but it is not possible in every case.

2.5 Ain (ع)

Ain (ع – glottal stop) exists in the Arabic alphabet and native Arabic speakers pronounce it properly. Urdu also has adopted Ain (ع) in its alphabet as well as Arabic loan words but native speakers of the sub-continent cannot produce its sound properly, rather they produce a vowel sound by replacing Ain (ع) with Alef (ا). The Hindi alphabet follows one character for one sound rule and it does not have any equivalent of Ain (ع). Then, Ain (ع) in Urdu words is transcribed in Hindi by some vowel representing the pronunciation of the word by native subcontinent speakers. Thus it is always transliterated in some vowel in Hindi. For example, Ain (ع) gives the sound of the vowel [ə] in عجیب – अजीब (strange) [ədʒib] and the vowel [ɑ] with and without Alef (ا) in words عام – आम (common) [ɑm] and بعد – बाद (after) [bɑd̪] respectively. In some words, Ain (ع) is not pronounced at all and should be deleted while transliterating from Urdu to Hindi, *e.g.*, شُرُوع – शुरू (to start) [ʃʊru], *etc*. Conversion of Ain (ع) is a big problem for transliteration.

2.6 Nasalization

Noonghunna (ں) [ɲ] is the nasalization marker of vowels in Urdu. Interestingly, it is only used to nasalize a vowel at the end of a word. In the middle of a word, Noon (ن) [n] is used to mark the nasalization of a vowel and it is also used as a consonant. It is difficult to differentiate between nasalized and consonant Noon (ن). There are certain contextual rules that help to decide that Noon (ن) is used as a consonant or a nasalization marker, but it not possible in all cases.

2.7 Persio-Arabic Vocabulary

Urdu borrows a considerable portion of it vocabulary from Persian and Arabic and transliteration of these words in Hindi is not regular. Table 5 explains it with few examples.

Urdu	Hindi	
	FST Conversion	Correct
بالکل	बालकुल (surely)	बिलकुल [bɪlkʊl]
بالواسطہ	बालवासता (with reference of)	बिलवासता [bɪlvaṣ̌tə]
فی الحقیقت	फ़ीअलहक़ीक़त (in fact)	फ़िलहक़ीक़त [fɪlhəqiqət]

Table 5: Persio-Arabic Vocabulary in Urdu

3 Hybrid Transliteration Model

The analysis of the previous section clearly shows that solution of these problems is beyond the scope of the non-probabilistic Hindi Urdu Finite-state transliteration model (Malik *et al*., 2008). We propose a hybrid transliteration model that takes the input Urdu text and converts it in Hindi using the Finite-state Transliteration Model (Malik et al, 2008). After that, it tries to correct the orthographic errors in the transducer-only Hindi output string using a *statistical word language model* for Hindi with the help of a *Hindi Word Map* described later. The approach used is rather similar to what is done in text recapitalization (Stolcke *et al.* 1998) for instance.

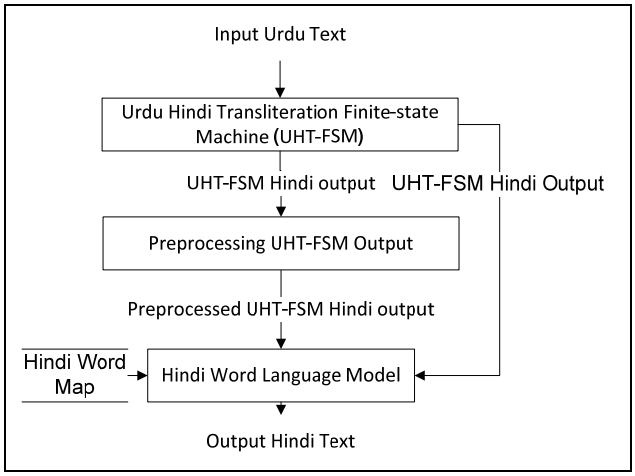

Figure 2: Hybrid Transliteration Model for Urdu Hindi

Normally, the Urdu text does not contain necessary diacritical marks that are mandatory for the correct transliteration by the finite-state component `Urdu Hindi Transliteration`

Finite-state Machine (UHT-FSM), described by Malik *et al.* (2008). The proposed hybrid model focuses on the correct transliteration of Urdu texts without diacritical marks. Figure 2 gives the proposed Model architecture.

3.1 Preprocessing UHT-FSM Output

The goal of this pre-processing is to generate a more "normalized" (and consequently more ambiguous) form of Hindi, e.g. pre-processing transforms both corpus words इस (this) [ɪs] and उस (that) [ʊs] (if encountered in the UHT-FSM Hindi output) into the default input Hindi word अस* [əs] (not a valid Hindi word but is a finite-state transliteration of the input Urdu word اس, a word without diacritical marks). Thus pre-processing is vital for establishing connections between the UHT-FSM Hindi output words (from the Urdu input without diacritical marks) and the Hindi corpus words. In the example above, the word अस* [əs] is aligned to two Hindi corpus words. All such alignments are recorded in the *Hindi Word Map*. This ambiguity will be solved by the Hindi word language model, trained on a large amount of Hindi data. Thus pre-processing is a process that establishes connections between the most likely expected input Hindi word forms (UHT-FSM Hindi output from the Urdu input without diacritical marks) and the correct Hindi word forms (words that are present in the Hindi corpus).

The Preprocessing component is a finite-state transducer that normalizes the Hindi output of UHT-FSM component for the Hindi word language model. The transducer converts all cases of gemination of consonants into a simple consonant. For example, the UHT-FSM converts the Urdu word ربّ (God) [rəbb] into रब्ब and the Preprocessing converts it into रब [rb]. The transducer also removes the conjunct marker (्) from the output of the UHT-FSM except when it is preceded by one of the consonant from the set {र [r], ल [l], म [m], न [n]} and also followed by the consonant ह [h] (first 3 lines of Figure 3), *e.g.*, UHT-FSM converts the Urdu words ہِندی (Hindi) [hɪndi] and دُلہن (bride) [d̪ʊlʰn] into हिन्दी and दुल्हन respectively and the Preprocessing component converts them into हिनदी (removes ्) and दुल्हन (no change). Actually, Preprocessing deteriorates the accuracy of the output of the UHT-FSM component. We will come back to this point with exact figures in the next section.

The code of the finite-state transducer is given in XFST (Beesley and Karttunen, 2003) style in Figure 3. In XFST, the rules are applied in reverse order due to XFST's transducer stack, i.e. a rule written at the end of the XFST script file will apply first and so on.

```
read regex [् -> 0 || [? - [र | ल | म | न]] _ [? - ह]];
read regex [् -> 0 || [र | ल | म | न] _ [? - ह]];
read regex [् -> 0 || [? - [र | ल | म | न]] _ [ह]];
read regex [[क ् क] -> क, [क ् ख] -> ख, [ग ् ग] -> ग, [ग ् घ] -> घ, [च ् च] -> च, [च ् छ] -> छ, [ज ् ज] -> ज, [ज ् झ] -> झ, [ट ् ट] -> ट, [ट ् ठ] -> ठ, [ड ् ड] -> ड, [ड ् ढ] -> ढ, [त ् त] -> त, [त ् थ] -> थ, [द ् द] -> द, [द ् ध] -> ध, [प ् प] -> प, [प ् फ] -> फ, [ब ् ब] -> ब, [ब ् भ] -> भ, [म ् म] -> म, [य ् य] -> य, [र ् र] -> र, [ल ् ल] -> ल, [व ् व] -> व, [श ् श] -> श, [ष ् ष] -> ष, [स ् स] -> स, [ह ् ह] -> ह, [क़ ् क़] -> क़, [ख़ ् ख़] -> ख़, [ग़ ् ग़] -> ग़, [ज़ ् ज़] -> ज़, [ड़ ् ड़] -> ड़, [ढ़ ् ढ़] -> ढ़, [फ़ ् फ़] -> फ़];
```

Figure 3: Preprocessing Transducer

3.2 Hindi Word Language Model

The Hindi Word Language Model is an important component of the hybrid transliteration model. For the development of our *statistical word language model*, we have used the Hindi Corpus freely available from the Center for Indian Language Technology[1], Indian Institute of Technology Bombay (IITB), India.

First, we extracted all Hindi sentences from the Hindi corpus. Then we removed all punctuation marks from each sentence. Finally, we added '<s>' and '</s>' tags at the start and at the end of each sentence. We trained a tri-gram Hindi Word Language Model with the SRILM (Stolcke, 2002) tool. The processed Hindi corpus data contains total 173,087 unique sen-

[1] http://www.cfilt.iitb.ac.in/

tences and more than 3.5 million words. The SRILM toolkit command '*disambig*' is used to generate the final Hindi output using the *statistical word language model* for Hindi and the *Hindi Word Map* described in the next section.

3.3 Hindi Word Map

The *Hindi Word Map* is another very important component of the proposed hybrid transliteration model. It describes how each "normalized" Hindi word that can be seen after the `Preprocessing` step and can be converted to one or several correct Hindi words, the final decision being made by the *statistical word language model* for Hindi. We have developed it from the same processed Hindi corpus data that was used to build the `Hindi Word Language Model`. We extracted all unique Hindi words (120,538 unique words in total).

The hybrid transliteration model is an effort to correctly transliterate the input Urdu text without diacritical marks in Hindi. Thus we take each unique Hindi word and try to generate all possible Hindi word options that can be given as input to the `Hindi Word Language Model` component for the said word. Consider the Urdu word ربّ (God) [rəbb]; its correct Hindi spellings are रब्ब. If we remove the diacritical mark Shadda (ّ) after the last character of the word, then the word becomes رب and `UHT-FSM` transliterates it in रब*. Thus the `Hindi Word Language Model` will encounter either रब्ब or रब* for the Hindi word रब्ब (two possible word options). In other words, the *Hindi Word Map* is a computational model that records all possible alignments between the "normalized" or pre-processed words (most likely input word forms) and the correct Hindi words from the corpus.

We have applied a finite-state transducer that generates all possible word options for each unique Hindi word. We cannot give the full XFST code of the 'Default Input Creator' due to space shortage, but a sample XFST code is given in Figure 4. If the Urdu input contains all necessary diacritical marks, then pre-processing of the output of the `UHT-FSM` tries to remove the effect of some of these diacritical marks from the Hindi output. In the next section, we will show that actually it increases the accuracy at the end.

```
define CONSONANTS [क | ख | ग | घ | ङ | च | छ | ज | झ | ञ | ट | ठ | ड | ढ | ण | त | थ | द | ध | न | प | फ | ब | भ | म | य | र | ल | व | श | ष | स | ह | क़ | ख़ | ग़ | ज़ | ड़ | ढ़ | फ़];
…
read regex [ै (->) े, ी (->) े, ू (->) ो, ौ (->) ो, ि (->) 0, ु (->) 0 || [CONSONANTS] _ ];
read regex [ी (->) े || [CONSONANTS] _ [? - .#.]];
read regex [ि -> ी, ु -> ो, ु -> ू || [CONSONANTS] _ .#.];
…
```

Figure 4: Default Input Creator Transducer

Practically, the *Hindi Word Map* is a file in which each line contains a possible input word to `Hindi Word Language Model`, followed by a list of one (see line 3 of Figure 5) or more (see line 1 of Figure 5) words from the corpus that are associated with this possible input word.

The 'Default Input Creator' transducer has generated in total 961,802 possible input words for 120,538 unique Hindi words. For implementation reasons, we also added non-ambiguous pair entries in the word map (see line 2 of Figure 5), thus the initial word map contains in total 1,082,340 entries. We extract unique option words and finally, *Hindi Word Map* contains in total 962,893 entries. Some examples from Hindi Word Map file are given in Table 5.

```
(1) कीजे कीजि कीजै
(2) कीजो कीजो
(3) रब रब्ब
(4) कीमयागरी कीमियागरी कीमियागिरी
(5) अस इस उस
```

Figure 5: Sample Hindi Word Map

4 Test and Results

For testing purposes, we extracted 200 Hindi sentences from the Hindi corpus before removing punctuation marks. These sentences were of course removed from the training corpus used to build the *statistical word language model* for Hindi. First we converted these 200 Hindi sentences in Urdu using Hindi Urdu Finite-state transliteration model (Malik *et al*., 2008). Trans-

literated Urdu sentences were post edited manually for any error and we also made sure that the Urdu text contained all diacritical marks. 200 original Hindi sentences served as Hindi reference for evaluation purposes.

From the post-edited Urdu sentences, we developed two test corpora. The first test corpus was the Urdu test with all diacritical marks. In the second test corpus, all diacritical marks were removed. We calculated both word level and character level accuracy and error rates using the SCLITE[2] tool. Our 200 sentence test contains 4,250 words and 16,677 characters in total.

4.1 Test: UHT-FSM

First we converted both Urdu test data using `UHT-FSM` only and compared the transliterated Hindi texts with the Hindi reference. `UHT-FSM` shows a word error rate of 21.5% and 51.5% for the Urdu test data with and without diacritics respectively. Results are given in Table 6, row 1.

Urdu Test Data	With diacritics	Without diacritics
UHT-FSM Accuracy/Error	80.7% / 21.5%	50.7% / 51.5%
UHT-FSM + HLM	82.6% / 19.6%	79.1% / 23.1%
UHT-FSM + PrePro	67.5% / 32.4%	50.7% / 51.5%
UHT-FSM + PrePro + HLM	85.8% / 16.4%	79.1% / 23.1%

Table 6: Word Level Results

These results support our claims that the absence of diacritical marks considerably increases the error rate.

4.2 Test: UHT-FSM + Hindi Language Model

Both outputs of `UHT-FSM` are first passed directly to `Hindi Word Language Model` without preprocessing. The `Hindi Word Language Model` converts UHT-FSM Hindi output in the final Hindi output with the help of *Hindi Word Map*.

Two final outputs were again compared with the Hindi reference and results are given in Table 6, row 2. For Urdu test data without diacritics, error rate decreased by 28.4% due to the `Hindi Word Language Model` and *Hindi Word Map* as compared to the `UHT-FSM` error rate. The Hindi Word Language Model also decreases the error rate by 1.9% for the Urdu test data with diacritics.

4.3 Test: UHT-FSM + Preprocessing

In this test, both outputs of `UHT-FSM` were preprocessed and the intermediate Hindi outputs were compared with the Hindi reference. Results are given in Table 6, row 3. After the comparison of results of row 1 and row 3, it is clear that preprocessing deteriorates the accuracy of Urdu test data with diacritics and does not have any effect on Urdu test data without diacritics.

4.4 Test: UHT-FSM + Preprocessing + Hindi Language Model

Preprocessed UHT-FSM Hindi outputs of the test of Section 4.3 were passed to the `Hindi Word Language Model` that produced final Hindi outputs with the help of the *Hindi Word Map*. Results are given in Table 6, row 4. They show that the `Hindi Word Language Model` increases the accuracy by 5.1% and 18.3% when compared with the accuracy of `UHT-FSM` and `UHT-FSM` + Preprocessing tests respectively, for the Urdu test data with diacritical marks.

For the Urdu test data without diacritical marks, the `Hindi Word Language Model` increases the accuracy rate by 28.3% in comparison to the accuracy of the `UHT-FSM` output (whether pre-processed or not).

4.5 Character Level Results

All outputs of tests of Sections 4.1, 4.2, 4.3 and 4.4 and the Hindi reference are processed to calculate the character level accuracy and error rates. Results are given in Table 7.

Urdu Test Data	With diacritics	Without diacritics
UHT-FSM	94.1% / 6.5%	77.5% / 22.6%
UHT-FSM + HLM	94.6% / 6.1%	89.8% / 10.7
UHT-FSM + PreP	87.5% / 13.0%	77.5% / 22.6
UHT-FSM + PreP + HLM	94.5% / 6.1%	89.8% / 10.7

Table 7: Character Level Results

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

4.6 Results and Examples

The `Hindi Word Language Model` increases the accuracy of Urdu Hindi transliteration, especially for the Urdu input without diacritical marks.

Consider the examples of Figure 7. Figure 1 is reproduced here by adding the Hindi transliteration of example sentences using the proposed hybrid transliteration model and Hindi reference.

(i) مَیں نے بہُت ادھِک کام نہیں کیا ہَے (ii) میں نے بہت ادھک کام نہیں کیا ہے	(1)
(i) मैं ने बहुत अधिक काम नहीं किया है (ii) मैं ने बहुत अधक काम नहें कया हे I have not done a lot of work	
Output of Hybrid Transliteration Model (i) मैं ने बहुत अधिक काम नहीं किया है (ii) मैं ने बहुत अधिक काम नहीं किया है	
Hindi Reference मैंने बहुत अधिक काम नहीं किया है	
(i) کیئنْدْریّہ سْطر پر بھی اَور راجْیہ سْطر پر بھی (ii) کیندریہ سطر پر بھی اور راجیہ سطر پر بھی	(2)
(i) केन्द्रीय स्तर पर भी और राज्य स्तर पर भी (ii) केंदरय सतर पर भी ओर राजय सतर पर भी Both at the central level and at the state level	
Output of Hybrid Transliteration Model (i) केन्द्रीय स्तर पर भी और राज्य स्तर पर भी (ii) केंद्रिय स्तर पर भी और राज्य स्तर पर भी	
Hindi Reference केन्द्रीय स्तर पर भी और राज्य स्तर पर भी	

Figure 7: Examples

By comparing Hindi outputs of `Hindi Word Language Model` with the Hindi reference, only the first word of (2)(ii) is wrong and other errors due to the absence of diacritical marks in the source Urdu sentences are corrected properly.

5 Conclusion

From the test results of the previous section we can conclude that the *statistical word language model* increases the accuracy of Urdu to Hindi transliteration, especially for Urdu input text without diacritical marks. The proposed Hybrid Transliteration Model improves the accuracy and produces the correct Hindi output even when the crucial information in the form of diacritical marks is absent. It increases the accuracy by 28.3% in comparison to our previous Finite-state Transliteration Model. This study also shows that diacritical marks are crucial and necessary for Hindi Urdu transliteration.

References

Beesley, Kenneth R. and Karttunen, Lauri. 2003. *Finite State Morphology*, CSLI Publication, USA.

Grimes, Barbara F. (ed). 2000. *Pakistan*, in Ethnologue: Languages of the World, 14th Edition Dallas, Texas; Summer Institute of Linguistics, pp: 588-598.

Hussain, Sarmad. 2004. *Letter to Sound Rules for Urdu Text to Speech System*, proceedings of Workshop on Computational Aproaches to Arabic Script-based Languages, COLING 2004, Geneva, Switzerland.

Jawaid, Bushra and Tafseer Ahmed. 2009. *Hindi to Urdu Conversion: Beyond Simple Transliteration,* in proceedings of Conference on Language & Technology, Lahore, Pakistan.

Kellogg, Rev. S. H. 1872. *A Grammar of Hindi Language,* Delhi, Oriental Book reprints.

Khan, Mehboob Alam. 1997. اُردو کا صوتی نظام (Sound System in Urdu), National Language Authority, Pakistan

Knight, K. and Graehl, J. 1998. *Machine Transliteration,* Computational Linguistics, 24(4).

Knight, K. and Stall, B. G. 1998. *Transliterating Names and Technical Terms in Arabic Text,* proceedings of COLING/ACL Workshop on Computational Approaches to Semitic Languages.

Malik, M. G. Abbas. Boitet, Christian. Bhattcharyya, Pushpak. 2008. *Hindi Urdu Machine Transliteration using Finite-state Transducers,* proceedings of COLING 2008, Manchester, UK.

Montaut, A. 2004. *A Linguistic Grammar of Hindi,* Studies in Indo-European Linguistics Series, Munchen, Lincom Europe.

Paola, V. and Sanjeev, K. 2003. *Transliteration of Proper Names in Cross-language Application,* proceedings of 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Toronto, Canada.

Pirkola, A. Toivonen, J. Keshustalo, H. Visala, K. and Jarvelin, K. 2003. *Fuzzy Translation of Cross-lingual Spelling Variants,* proceedings of 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Toronto, Canada.

Rahman, Tariq. 2004. *Language Policy and Localization in Pakistan: Proposal for a Paradigmatic*

Shift, Crossing the Digital Divide, SCALLA Conference on Computational Linguistics.

Stolcke, A. 2002. *SRILM – An Extensible Language Modeling Toolkit*, in proceedings of International Conference on Spoken Language Processing.

Stolcke, A. Shriberg, E. Bates, R. Ostendorf, M. Hakkani, D. Plauche, M. Tur, G. and Lu, Y. 1998. *Automatic Detection of Sentence Boundaries and Disfluencies based on Recognized Words*. Proceedings of International Conference on Spoken Language Processing (ICSLP), Sydney, Australia.

Yan, Qu. Gregory, Grefenstette. and David A. Evans. 2003. *Automatic Transliteration for Japanese-to-English Text Retrieval*. In proceedings of the 26th annual international ACM SIGIR conference on Research and Development in Information Retrieval, pp: 353 – 360.

Zia, Khaver. 1999. *Standard Code Table for Urdu.* Proceedings of 4th Symposium on Multilingual Information Processing (MILIT-4), Yangon, Myanmar, CICC, Japan.