Urdu-Hindi-Urdu Machine Translation: Some Problems

Amba Kulkarni
Rahmat Yousufzai
Pervez Ahmed Azmi

Department of Sanskrit Studies,
University of Hyderabad,
Hyderabad,
India
ambapradeep@gmail.com,
rahmat_yousufzai2001@yahoo.com

Abstract

In this paper we discuss the problems in Urdu-Hindi-Urdu Machine Translation at various levels. Though because of large common vocabulary it may sound that only transliteration can help to overcome the language barrier between Urdu and Hindi, the tendency of Urdu to use words from Persian and Arabic origin, and the tendency of Hindi to use words of Sanskrit origin, call for the use of proper Machine Translation System. However, we point out the problems at various levels of Machine Translation, and suggest an alternative approach. Following this alternative approach a working system has been built and is available at http://sanskrit.uohhyd.ernet.in/~anusaaraka/urdu/Urdu-Hindi-Translation.

1. Introduction:

Urdu and Hindi are very widely spoken languages in the world, particularly in the Indian subcontinent. Both have Indian origin and have drawn from Sanskrit through Shourseni, Apbhransh and Khadi Boli. The syntax of both languages is almost the same and there are many words and expressions commonly used in both the languages. The common language with common vocabulary is referred to as Hindustani that could be written in both the scripts that is Devanagari and Persio-Arabic. Use of two scripts for Hindustani has divided the world of Hindustani into two. Urdu has a tendency to use words from Persian and Arabic origin, whereas Hindi has a tendency to adopt words from Sanskrit. Thus Hindustani with more words from Persio-Arabic becomes Urdu while Hindustani with more words from Sanskrit becomes Hindi. During the Mughal Empire and years there after, lot of Persio-Arabic words have entered the common vocabulary of Hindi. This raises an important issue. The common vocabulary in Hindi and Urdu tempts a Urdu-Hindi-Urdu Machine Translation developer towards the transliteration. At the same time the presence of Persio-Arabic words in Urdu and Sanskrit words in Hindi, along with certain structural differences demand various modules such as Morphological Analyser, POS Tagger, Chunker, etc. to be part of a Machine Translation system. In this paper we discuss the problems at various levels of Machine Translation and finally suggest a model for developing an easy access of Urdu text through Hindi and vice versa.

2. Transliteration Module:

A large common vocabulary makes an Urdu-Hindi transliteration module an important component of MT system. Unlike majority of Indian scripts which originated through the Brahmi script, Urdu uses Persio-Arabic script. Urdu has 38 consonants while Hindi has 33 consonants which are part of Devanagari. Further Hindi has adopted few more consonants such as: क., ख., ग., ज.,ड. to represent faithfully Urdu consonants (ق ، خ، غ، ز، ڑ). These are generated typically by placing a nukta (.) character below these consonants. Urdu does not have special symbols for aspirated. An aspirated consonant is represented orthographically as a corresponding non-aspirated consonant followed by do-chashmi he (ھ). For example bha (भ) = Be (ب) + do chshmi he (ھ). Urdu does not have conjunction of consonants as in Hindi and thus there is no concept of halant in

Urdu alphabet. However to represent the conjunction the diacritic mark Jazam (◌ْ) is used. Hindi has vowels and vowel modifiers. Urdu on the other hand does not have any pure vowels except alif (ا). The semi vowels waw (و) , choti ye (ی) and badi ye (ے) along with alif (ا) play the role of long vowels when required. Urdu does not have any short vowels; instead, it has the diacritic marks zer(ِ), zabar (َ) , pesh (ُ), Jazam (ْ), tashdeed (ّ) and Tanveen (ً) which are used very rarely or only in the basic / elementary texts. The literary texts, news-papers and web sites, rarely use these marks leaving the text ambiguous.

2.1. Urdu-Hindi Transliteration:

The requirements of a good transliteration scheme among Urdu to Hindi then are:

a) words common to both Urdu & Hindi should be transliterated correctly as per their conventional spellings.

b) Persio-Arabic words in Urdu that are not common in Hindi should be transliterated to the phonetically closed spellings.

The problem of translation Urdu-Hindi then reduces to:

- Identifying consonant clusters as conjuncts,
- Identifying missing short vowels,
- Disambiguating the semi vowels waw (و), choti ye (ی) and badi ye (?),
- In addition there are less frequent occurrences of noon (ن) and noon-e-ghunna (ں), which need to be mapped to the corresponding nasalized consonants, similarly *he* (?) at the end need to be mapped to either ◌ा or ह, etc.

Followings are same of the possible approaches:

a)Have a good coverage Urdu-Hindi dictionary of common Hindustani words, written both in Urdu as well as Devanagari script. This approach definitely is the best one. However to start with, till such a dictionary be made available in electronic form, one needs to have an alternative approach.

b)Have a good coverage Hindi Monolingual dictionary. The transliterated word from Urdu will be searched in this dictionary for the best match and all possible answers will be returned. If Hindi lexicon is available with frequency distribution data, then the frequency information will be used to prune out less frequent matches.

c) The above resources may also be used to try various Machine Learning techniques.

Frequency distribution of Hindi words (CIIL corpus) was available readily and hence we followed the approach (b) and the results are summarized as follows.

Urdu text	Size in words	Correct Transliteration (%)
Tourism text1	824	98.4%
Tourism text2	666	99.1%
Health Text	334	95%

2.2. Hindi-Urdu Transliteration:

The problem of Hindi to Urdu transliteration is easier on account of the following:

- The conjuncts in Hindi need to be split as sequence of full consonants or in other words the additional '*halant*' character in Devanagari needs to be deleted.
- Vowels get mapped to the corresponding diacritical marks and may be dropped easily if not required.
- The long vowels are mapped to either waw (و) or ye (? ی) according to the Panini's rule स्थान s नतरतम (Panini:1.1.50). The one which is the closest with respect to the place of articulation, is the best match.

The major issues then are:

- Though Hindi has extended the Devanagari script by adopting the nukta character and coining new consonants with this nukta, there is no uniformity among the Hindi users in the use of these adapted consonants. This then leads to wrong Urdu spelling in the transliteration.
- The missing consonants in Hindi also introduce some errors. However since the words that use (ض ص) are basically of persio-Arabic origin and not used frequently in Hindi, the transliteration from Hindi to Urdu as far as these consonants are concerned, does not pose much problem.
- It is the alif (ا) and ain (ع) which are the major trouble-givers. Unless one refers to the dictionary, the correct spelling can't be guessed in such case. To handle this

ambiguity, we use the Urdu-Hindi bilingual dictionary.

Following table shows the performance of the current system using the above mentioned rules.

Hindi text	Size in words	Correct Transliteration (%)
text1	381	95.6%
text2	415	95.7%
text3	482	97.6%

3. Morphological Analyzer :

The Finite State Transducer approach to morphology has became very common and popular among the developers of morphological analyzers and generators. In the past decade one will see up-shoot of Morph Analyzer for a variety of languages like European, Indian, Arabic, etc. Since Urdu borrows heavily from Hindi as well as Persian and Arabic, it has a mixed morphology. The morphology of Hindi is very simple and can be best captured by the word and paradigm model (Bharati, 1995).

The morphology for the Persio-Arabic words on the other hand is an item and process based. Simple word paradigm model is not sufficient since the orthography does not reflect the underlying vowel combination. In case of Persian and Arabic languages it is the vowel combinations which determine the paradigm. Thus as tried by Beesley(1998) for Arabic, a two level analysis - one representing the combinations of consonants in the roots and the other representing the vowel combinations is required. Since Urdu is spoken at various parts of India that are linguistically surrounded by other Indian languages, the Persio-Arabic words are treated like “borrowed” words and inflected according to the rules of Hindi morphology also. Thus in the Urdu spoken in India we encounter words such as (مکانات) as well as (***). This makes the Urdu morphology more complex. Urdu does not have Persio-Arabic verbs and hence the verb morphology for Urdu is same as that of Hindi.
We have built a Morph Analizer based on the word and paradigm approach. Because of the complexity as mentioned above the number of noun paradigms is 155 as against 34 in Hindi. The appendix lists the total paradigms as well as the default paradigms in each case, for both nouns as well as verbs.

We have built this analyzer with an off the shelf FST (available under GPL at http://www.apertium.org) and tested on a vocabulary of around 13,000 noun root entries. The performance is 95% for verbs. But for nouns it was found to be only 60%. Major problems were because of non-availability of root words in the dictionary, and not because of any missing paradigms. Unlike Hindi or any other Indian Language, it was little difficult in case of Urdu to decide the default paradigm. In Indian Language the words are marked with vowels and the vowels at the end of a word decide the paradigm. However since in Urdu, the orthography does not mark the vowel, it was difficult to decide the default paradigm. We used the dictionary of pronunciation which contain the missing vowels to decide the paradigm.

4. Standardization Issues:

Urdu data entry operators do not enter the data in a standardized format which creates problems in transliteration as well as it increases the ambiguity. The problems in e-representation of Urdu texts may be classified into three categories:

a) wrong spellings:

- ی and ? when in middle is not differentiated. ?ل and میل are written in the same way (میل). Many a times ? is written ی and ی is written as ? due to the same appearance in Urdu text editors.
- ں in middle is written as ن making the transliteration difficult.

b) rare use of diacritic marks:

Diacritic marks Zabar, Zer, Pesh, Tashdeed, Jazam are normally not written hence differentiation between اِس and اُس is difficult.

c) wrong word splittings:

- In Urdu, there are certain characters which do not join with forthcoming characters like ا ، د ، ذ ، ڈ ، ر ، ز ، ژ ، ژ etc. and the operator does not give space after the completion of the word resulting the coalition of two words.
- There are also certain instances where the words are split since otherwise the shape of the characters change when they are in between. For example جائں گی ،جائں گی، بی بی ، پس منظر

From Machine Translation point of view, these are crucial issues, since otherwise the performance of the system goes down drastically. We have noticed that there are around 5% such errors. This is very significant when we compare it with the transliteration errors which are less than 5%.

5. Need of a New Architecture for MT:

In the conventional Machine Translation approach, the modules are serially connected to each other. This means, the errors get cascaded. Among all the modules, one can guarantee theoretically, 100% reliability only for a morphological analyzer. Of course, in practice, the presence of proper nouns, in the absence of a good quality Named Entity Recogniser for languages which do not posses any special mark, as in the case of English which uses capital letters, the performance goes down. The next module viz. POS tagger takes output of morphological analyzer as an input and proposes the most likely POS tagger which helps one to prune out all less likely morphological analysis in that context. The state of art performance of POS tagger for Indian Languages is not more than 90%. The POS taggers of Urdu developed in-house using the Markov model gives a performance of 85%. The state of art performance of Word Sense Disambiguator modules is far below any acceptable range. Its performance decreases further because of the erroneous input of earlier modules. Since both Urdu and Hindi have many common words, the ambiguity gets carried over from one language to the other. So it is not necessary to disambiguate these words. For example the Urdu word پر has two meanings. As a disjunct operator it means कि·तु and as a noun it means प·ख. Hence it is not at all necessary to disambiguate the word for its POS tagger and then for its meaning. One can directly map this Urdu word پر to the corresponding Hindi word पर. Among the 10,000 high frequency words, only 217 have multiple mappings from Urdu-Hindi. That means the percentage of words whose ambiguity needs to be resolved is only 2. Thus unless we have a POS tagger which performs better than 98%, we can not have better quality output in the Urdu-Hindi Machine Translation system. Same is the case with the Word Sense disambiguator. It is not at all necessary to call WSD module for the words where the ambiguity gets carried over. We describe an alternative approach which takes advantage of the closeness of the two languages both at the lexical level as well as at the syntactic level.

6. Alternative Approach:

We have seen in the beginning that Urdu and Hindi share a common vocabulary of Hindustani. So these common words need to be just transliterated, since they carry the ambiguity if any, to the other language as well. Sometimes, the orthographic differences between the two languages also create an extra ambiguity. The words from Persian or Arabic origin in case of Urdu and the words from Sanskrit origin in case of Hindi, which are not commonly used in Hindustani need to be translated. Thus we require two different treatments for two different sources of words. Here under we describe the alternative approach for Urdu-Hindi Machine Translation which can then be easily adapted for the Hindi-Urdu Machine Translation as well.

Our default assumption is that the word is from Hindustani. The Arabic and Persian words are treated as exceptions. So we build a special morphological analyser for Arabic and Persian words only. We pass the words through this morphological analyser. All the recognised words are searched in the bilingual dictionary and mapped and generated into Hindi. The un-recognised words, as per our default assumption are the Hindustani words. We first check them for orthographic ambiguities if any. We resolve these ambiguities first by using a simple collocation based word sense disambiguator developed in-house. All the remaining words are transliterated into Devanagari. Since the disambiguator may fail, we also provide alternate meanings in the tool-tip, along with the original Urdu sentence. This helps both the reader as well as the developer to fix the errors. This system is available at http://sanskrit.uohyd.ernet.in/~anusaaraka/urdu/Urdu-Hindi-Translation/ for use.

7. Conclusion:

The development of several language analysis tools such as morphological analyser, generator, POS tagger, Chunker, Parser, is very important for any Machine Translation activity. But at the same time depending heavily on these and strictly following a particular approach may delay the development of Machine Translation systems for the languages which are very close to each other both syntactically as well as semantically. In the last few decades we have seen the growth of good quality Machine translation systems among the European languages. It should be possible to develop good quality Machine translation systems for Indian languages by adapting alternative approaches very quickly. The availability of such systems also helps in reducing the divide between Urdu-Hindi, which is merely because of the scripts. The electronic media can help in bridging the divide should we follow a right approach. In this paper we have illustrated the development of such a system in short for Urdu-Hindi.

8. Acknowledgment:

This work is supported financially by the Ministry of Information Technology, Government of India, under the Indian Language to Indian Language Machine Translation Consortium project, 2006-2008.

9. References:

Bharati, Akshar, Vineet Chaitanya, Rajeev Sangal , *Natural Language Processing: A Paninian Perspective*, Prentice-Hall of India, 1995.

Beesley, K. R., *Arabic Morphological Analysis on the Internet*, Proceedings of the International Conference on Multi-Lingual Computing, 1998

CIIL corpus: http://www.ciil.org