IMPROVING ENGLISH TO URDU MACHINE TRANSLATION VIA TRANSLITERATION

M. PRANEESH
Assistant Professor,
Department of Computer Science
Sri Ramakrishna College of Arts and Science,
Coimbatore, India.
raja.praneesh@gmail.com

D. NAPOLEON
Assistant Professor,
School of Computer Science and Engineering,
Bharathiar University,
Coimbatore India
mekaranapoleon@yahoo.co.in

Abstract— **In this paper, a deep inspection on the integration of a transliteration system into general purpose English to Urdu phrasebased statistical machine translation system is given. We check on the integration from different guise and grade the improvement that can be attributed to the integration using BLEU automatic machine evaluation metric. From our experiment results it is concluded that a transliteration module can help in the situation where data sparsity exists.**

Keywords — *Parallel Corpus, Natural language Processing (NLP), Phrase-based translation, Statistical Machine Translation (SMT), Transliteration.).*

I. INTRODUCTION

IN this section, we discussed the actual process of transliteration that can be used for getting improved machine translation (MT) system for different languages. An overview of MT with SMT is also discussed briefly.

A. *Machine Transliteration*

Method of copying the words from one script to phonetically counterpart words in another script is known as transliteration. There are rules which purvey mapping from the letters of source script alphabet to the letter of target script alphabet but this based on phonetic similitude. Translated dictionaries are not wide ranging and inoperative for proper noun e.g. people, places, companies therefore this process is highly effective for transliteration. There are lot of English text along with Urdu text on the websites and user interfaces etc. this text is created by applications such as screen reader, web page reader etc. and then this is forward to Urdu language processing application like machine translation or text to speech system for speech generation. English text is riddance by Urdu TTS or machine translation system and resulted translations is not enough consistent. Transliteration is the process of representing words from one language using the approximate phonetic or spelling equivalents of another language (Arbabi et al. 1994). In tasks such as MT and CLIR, the lack of a comprehensive bilingual dictionary including the entries for all entity names makes machine transliteration necessary (Kashani et al. 2007). The main motivation for integrating a machine transliteration module in NLP applications is to handle unseen terms in a proper way so that performance of that application is improved.

For most part of transliteration from Urdu to English is tough work because there is a difference in the phonological and orthographic systems of the two languages. On the one hand,

There are cases where an Urdu letter can be pronounced in multiple ways. For example, Urdu word " ا "can be pronounced

either as [a] or as [e]. On the other hand, two different Urdu sounds can be mapped into the same English letter. For example, both دand ڈare in most cases mapped to [d]. A prime obstacle dams from the fact that in the Urdu orthography like Arabic words are presented as arrays of constants where vowels are partially and very fickly represented. Sometime letters that are conjecture as representing vowels may even represent consonants, [v]/ [o]/ [u] and [y]/ [i]. As a result, the mapping between Urdu orthography and phonology is highly equivocal. Transliteration has procured a growing interest recently, peculiarly in the field of Machine Translation (MT). It drives those conditions where no translation would fulfill or even exist. If someone wants transliteration in the context of MT system, one has to recognize which terms should be transliterated instead of translated and then apply an exact transliteration for these terms. We focus on both chores in this work. Identification of Terms To-be Transliterated (TTT) must not be confused with recognition of Named Entities (NE) (Hermjakob et al., 2008).
On the one hand, many NEs should be translated rather than transliterated, for example:
Urdu has adopted many foreign words; it is encouraged to translate these in local language if the Urdu variation is available.
For example:
Museum <>میوزیم vsعجا ئب گھر
Government <>گو رنمنٹ vsحکو مت
Whether to translate or transliterate proper nouns such as:
New England Journal of Medicine
نیو انگلینڈ جرنل آف میڈیس
نیوانگلینڈ رسا لہء طب
What about "Apple Computer"?
ایپل یوٹر
سیب کمپیوٹر
Transliteration as a self-contained task has many challenges but when it applies in different real applications, it creates many new challenges. Analysis of effectiveness of

integrating a transliteration module in to a real MT system and evaluation the performance and accomplishment is the primary purpose of this paper.

B. *Machine Translation*

The Machine Translation (MT) can be defined as an automated system that analyses text from a Source Language (SL), apply some computation on that input and produces equivalent text in a required target language (TL) ideally without any kind of human intervention. There are two different approaches to resolve the problems of MT. One is the rule-based approach and other corpus-based approach. In the rule-based approach, the text in the source language is analyzed using various tools such as: a morphological parser and Analyzer and are transformed into an intermediate representation. Certain rules are used to generate the text in target language of this intermediate representation. A large number of rules are necessary to capture the phenomena of natural language. These transfer rules transfer the grammatical structure of the source language into target language. As the growth of rules increases, the system becomes very complicated. Formulation of a large number of rules is a tedious process. In statistical approaches, target text is generated and scored through a statistical model, the parameters of which are learned from parallel corpus. Here, MT is also seen as a decision problem, a better target language phrase id is decided from the given source language. Bayes rule and statistical decision theory are used to solve this decision problem. Statistical decision theory and Bayesian decision rules are used to minimize errors of decision. Statistical machine translation (SMT) gives better results if additional training data are available. SMT is superior to rule-based and example-based systems in that they do not require human interpenetration and can build a translation system in an unsupervised manner directly from the training data.

II. PREVIOUS WORK

Some research work has been done to translate Indian languages, mostly focusing Hindi and Bengali. Statistical Machine Translation (SMT) models have also been adapted or machine transliteration. (Matthews,.2007) for example uses Moses, a state-of-the-art Phrase-based Statistical Machine Translation system (PSMT) on transliterating proper names between English and Chinese, and English and Arabic. The parallel corpus from which the translation model is acquired contains approximately 2500 pairs, which are part of a bilingual person names corpus (LDC2005G02). This biases the model toward transliterating person names. The language model presented for that method consisted of 10K entries of names which are, again, not complete. It shows that a machine transliteration system can be built from an existing PSMT system whose performance is comparable to state-of-the-art systems designed specifically to transliterate. Our work is also based on such scenario. Although a lot of work has been done for Machine transliteration, little has been done with regard to translating transliterated entity names whose origin is in a different writing system. For maximum phrase length n the translation model it applies different settings and different n-gram order for the language model. In Arabic to English transliteration accuracy is 43%. The classification is based on pair wise features: sets of substrings are extracted from each of the words, and substrings from the two sets are then coupled to form the features. The accuracy of rightly recognized transliteration pair in top1 is 52% and in top 5 it was 88%. This method selects most accurate transliteration out of a list of candidates; our approach generates a list of possible transliterations positioned by their accuracy. (Stalls & Knight, 1998) propose a method for back transliteration of names that originate in English and occur in Arabic texts. The process uses a variety of probabilistic models for conversion of names written in Arabic into the English text. First of all, an Arabic name forward through a phonemic structure causing a network of possible English sound sequences where each sounds' probability is location dependent. After those phonetic sequences convert into English phrases. In the final stage each expected result is scored according to a unigram word model. 32% of the names are correctly translated by this method. Those which are not translated are mostly not foreign names. Pronunciation dictionary is used in this method therefore unable to transliterating only words of known pronunciation. Only one directional transliteration is performed by the above two methods i.e., either forward or backward transliteration while our work handles both. (Al-Onaizan & Knight, 2002) describe a system which combines a phonetic based model with a spelling model for transliteration. The spelling-based model directly maps sequences of English Letters into sequences of Arabic letters without the need of English pronunciation. The method uses a translation model based on IBM Model 1 (Brown et al., 1993), in which translation candidates of a phrase are generated by combining translations and transliterations of the phrase components, and matching the result against a large corpus. For top 10 results accuracy of overall system is 72% while for top 20 results accuracy is up to 84%. This method acts well for person names but it is restricted to transliterating NEs. As mention above TTT problem is different from NER problem. List of transliteration pairs from which transliteration model could be learned is basic requirement of this method. (Yoon et al. 2007) use phonetic distinctive features and phonology-based pseudo features to learn both language specific and language universal transliteration characteristics. Distinctive features are the attributes that determine the set of phonemic segments (consonants, vowels) in a given language. These distinctive features and so-called features are whip out from source and target language training data to train a linear classifier. The classifier reckons connectivity scores between English source words and target language words. The one with the highest score is selected as the transliteration when several target-language strings are transliteration candidates for a source word. With each of four target languages the method was considered using parallel corpora of English. NEs were extracted from the English side and were compared with all the words in the target language to find proper transliterations. The baseline presented for the case of transliteration from English to Arabic achieves Mean Reciprocal Rank (MRR) of 0.66 and this method improves its results by 7%. This technique involves knowledge about phonological characteristics, such as elision of consonants based on their position in the word, which requires expert knowledge of the language. In addition, transformation of stipulations into a phonemic representation poses obstacle in representing short vowels in Arabic and will have same behavior in Hebrew. English to Arabic transliteration is easier than Arabic to English, because in the former, vowels should be deleted whereas in the latter they should be generated. (Hermjakob et al. 2008) describe a method for identifying NEs that should be transliterated in Arabic texts. The method first tries to find a matching English word for each Arabic word in a parallel corpus, and tag the Arabic words as either names or non-names based on a matching algorithm. This algorithm implements a scoring model which applies manually-elaborate costs to pairs of Arabic and English substrings, permitting for context restraining. A number of language specific heuristics, such as reckon only capitalized words as candidates and using lists of stop words, are considered to highlight the algorithm's precision. The tagged Arabic corpus is then split: One part is used to collect statistics about the distribution of name/non-name patterns among tokens, bigrams and trigrams. The rest of the tagged corpus is used for training using an averaged perception. The precision of the identification task is 92.1% and its recall is 95.9%. This work also presents a novel transliteration model, which is integrated into a machine translation system. Its accuracy, measured by the percentage of correctly translated names, is 89.7%. (Khan, et al., 2013) presented baseline SMT system for English to Urdu translation using Hierarchical Model given by (Chiang, et al., 2005) They also made a comparison of simple default phrase-based model with the hierarchical model and showed the performance of simple phrase-based is much better for such local language like Urdu then the hierarchical phrase-based approach to SMT. (Sajjad et al. 2011) proposed a heuristic-based unsupervised transliteration mining system. It is the only unsupervised mining system that was evaluated on the NEWS10 dataset up until now, as far as we know. That system is computationally expensive.

III. Evaluation

In this section, we discuss translation and transliteration datasets used in our experiments then training, tuning and testing of different model components followed by the final results discussion.

A. Dataset

For this work, parallel corpora from more than one different domain were collected for translation and transliteration systems. The description of data collection, data resources and data processing is discussed in detail in this section. For transliteration the name entity bilingual corpus is extracted from web. It composed of around 15k names from around the world. This corpus is used for training tuning and testing of the transliteration system. The target side corpus of Urdu is used as monolingual data to train the language model for transliteration. This corpus is designed as one single word is divided into characters for both the source and target language. The characters are then acting as words and that actual word acts as a sentence while transliterating the source word. Some of the examples are given in Table 1.

TABLE 1
EXAMPLE FROM TRANSLITERATION CORPUS

Actual Source Word	*Converted Source Word*	*Actual Target Word*	*Converted Target Word*
ا سلم	ا س ل م	*Aslam*	*A s l a m*
محمد	م ح م د	*Muhammad*	*M u h a m m a d*
ا قرا	ا ق ر ا	*Iqra*	*I q r a*
راحیل	ر ا ح ی ل	*Raheel*	*R a h e e l*
ہا لینڈ	ہ ا ل ی ژ ن ڈ	*Holland*	*H o l l a n d*

For the bilingual corpus collection for translation our first motive was to collect data from as different domains as possible to get better translation quality and a wide range vocabulary. For this purpose, the corpus we selected to use in our work is EMILLE (Enabling Minority Language Engineering). EMILLE is a 63-million-word corpus of Indic languages (Baker, et al., LREC 2002) which is distributed by the European Language Resources Association (ELRA). EMILLE contains data from six different categories: consumer, education, health, housing, legal and social documents. This data is based on the information leaflets provided by the UK government and the various local authorities. We were provided in total 72 parallel files with each filename consisting of language code, text type (written or spoken), genre and subcategory, connected with hyphen character. The data is encoded in full 2-byte Unicode format and marked up in SGML format. We used this EMILLE corpus that is becoming a standard data repository for languages of this region. The parallel corpus consists of 200,000 words of text in English and its accompanying translations in Hindi, Bengali, Punjabi, Gujarati and Urdu. Its bilingual resources consist of roughly 12k sentences for all the available languages from which we were able to sentence-aligned and extract over 8k sentence for Urdu pairing with English using the sentence alignment algorithm given by (Moore.,2002).

TABLE 2
TRAINING AND EVALUATION DATA FOR EMILLE

Total Sentences	Training	Tuning	Testing
8200	6600	800	800

Cleaning of this corpus for making completely parallel and sentence aligned pair is very first step and also the toughest task that can be used for the development of any SMT system. A further detail about parallel data and vocabulary size is given in Table 2 & Table 3 respectively.

TABLE 3
EMILLE VOCABULARY SIZE

Source Language	***Target Language English***			
	Training Size (Tokens)		*Test Size (Tokens)*	
	Source	Target	Source	Target
Urdu	125,735	96,563	14,465	9.322

Large amount of Monolingual Urdu data that consists of flat sentences is collected for this research work. The monolingual corpus is used to make the language model that is used by the decoder to figure out which translation output is the most fluent among several possible translation options. In this study we also tried to gather huge monolingual data from as many different available online sources as possible. The next step is to train the language model on the corpus that is suitable to the domain. To fulfill this need, data from diverse domains is collected. The main categories of the collected data are, Religion, News, Education and numerous others. The target side of the parallel corpora is also added to the monolingual data. The monolingual corpora collected for this study have around 60 million tokens distributed in around 2 million sentences. These figures cumulatively present the statistics of all the domains whose data is used to build the language model. This statistic is after adding in the monolingual data the target side of all the parallel corpora we collected for this study.

B. *Experimental Setup*

For our gathered corpora, roughly 800 tuning and same number of sentences for testing along with above 6k sentences for training of the system All these statistics can be seen clearly in Table 2. After sampling of data, tokenization of training set, tuning set and test set is done for the dataset followed by lowercasing of dataset. All this is done by the scripts being supplied with the Moses decoder (Koehn et al., 2007). This lowercased training data is used for word alignment. Because of data sparsity we did not use cleaning script for filtering long sentences from the training data before training the system. **Baseline Settings for both translation & transliteration system:** We trained a Moses system (Koehn et al., 2007) with the following features: a maximum sentence length of 80, GDFA symmetrization of GIZA++ alignments (Och and Ney, 2003), an interpolated Kneser-Ney smoothed 5-gram language model with KenLM (Heafield,2011) used at runtime, a 5-gram OSM (Durrani et al., 2013), msd-bidirectional-fe, sparse lexical and domain features (Hasler et al., 2012), , 100-best translation options, MBR decoding (Kumar and Byrne, 2004), Cube Pruning (Huang and Chiang, 2007) with a stack-size of 1000 during tuning and 5000 during test, and the noreordering-over punctuation heuristic. No reordering is used for transliteration as we need the exact target word against the input source word so no kind of reordering is needed. We tuned with the k-best batch MIRA algorithm (Cherry and Foster, 2012). Language Model for translation system is built on the available monolingual Urdu corpus that we already discussed in last section. This language model is implemented as an n-gram model using the SRILM Toolkit.

Integration of transliteration system: After developing of both the translation and transliteration system using the Moses SMT toolkit with the given corpora. We did not evaluate the transliteration system separately as integration is being done as well. So the next step is to integrate and use the transliteration system to get some kind of fruitful output. For this integration we used the method given by (Durrani et al, 2014). First we run the translation system and get the output on our testing corpus for MT. After getting the output for that testing dataset we get the OOV from that output and manage a list of for those OOV words. This list is given as test set to the developed transliteration system. Now the OOV words in translation output will be replaced by the output transliterated words of transliteration system. This is the actual theme of integration of the transliteration system with the SMT system.

C. *Results*

All the evaluation scores and some sample translations from the developed SMT system are given in this section: As working on a very sparse resourced language, we have achieved much better BLEU scores. In Table 5 we presented the results of the experiments for both with transliteration and without transliteration. The results are composed of BLEU and NIST score evaluated over the test corpora that we discussed in previous section and also the count of Unknown words over test corpus. We got BLEU score of 0.1299 on developed SMT system. The BLEU score got some improvements when transliteration system is integrated with it. After the integration the BLEU score becomes 0.1308. Hence the transliteration system increases the BLEU score by less than 1%. For NIST we got 4.22 before and 4.24 after the integration of the transliteration system with pretty small amount of training parallel corpus. When counting the unknown words in translation of our SMT system we come up with 450 OOV words. For these words as we discussed above integrated the transliteration system to our translation system. The example given below shows the different kind of problems that are occurred in getting translation output from the developed system.

Example:
Source: Ahmad is a good boy.
Reference:احمد ایک اچھا لڑکا ہے
Output: Ahmad |0-0| ایک اچھا|1-2| لڑکا|3-3| ہے|4-4| ||

TABLE 4
URDU-ENGLISH PHRASE TABLE FOR GIVEN EXAMPLE

S. No	Input Phrase	Reference Phrase	Transliteration
1	Ahmad	NULL	احمد
2	good	ایک اچھا	آ گڈ
3	boy	لڑکا	بوائے
4	is	ہے	از

In output the first word "Ahmad" of source language English got no translation in output of target Urdu language. It decreases the actual BLEU score for this specific sentence. For this we use the integrated transliteration module and got the transliteration given in Table 4 for the first word "Ahmad". The next phrase got two target words in single phrase that can be seen in the phrase table entry. All this discussion with given output example lead us to a bottom line conclusion that if we manage to get a good tokenizer and also more corpora for this selected regional language. This can lead us to pretty much impressive BLEU score and a good fluent in order translation output much closer to reference translation.

TABLE 5
FINAL RESULTS BEFORE AND AFTER INTEGRATION OF TRANSLITERATION SYSTEM

Experiment	**BLEU**	**NIST**	**UNK**
Without Transliteration	0.1299	4.226	450
With Transliteration	0.1308	4.246	-

Translation quality can also be improved by studying the similarities between these languages. Data sparsity can be overcome by using methods of triangulation (trever, 2007) & (bertoldi, 2008) which have been shown to be useful for closely related languages.

IV. CONCLUSION & FUTURE WORK

The developed transliteration and translation system takes the English sentences as input and it generates corresponding best translation in Urdu. The translation of 800 sentences was evaluated using automatic evaluation metric i.e. BLEU evaluation. Average BLEU score of 10% to 15% was reported. The worth of the decoded sentences is directly be determined by the scope of the corpora and the excellence that specific corpora got. In future, we would like to study SMT by applying other different approaches to develop good language models and also the training model for some of the South Asian languages where more parallel corpus is available now or in nearer future.

References

[1] Anwar, 2009, a context-sensitive parser for Bangla-English machine translation, International Journal of Computer Science and Network Security, 09(08),317–326

[2] Baker P. [et al.] EMILLE, 2002, A 70-Million Word Corpus of Indic Languages: Data Collection, Mark-up and Harmonization: In Proceedings of the 3rd Language Resources and Evaluation Conference, pp. 819-825, LREC'.

[3] Cherry C,.et.al. 2012. Batch Tuning Strategies for Statistical Machine Translation. In Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 427–436, Montreal, Canada, June. Association for Computational Linguistics.

[4] Chiang D, 2005 , A hierarchical phrase-based model for statistical machine translation, Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics (ACL).

[5] Dasgupta S, et.al. 2004.An Optimal Way towards Machine Translation from English to Bengali, Proceedings of the 7th Inter-national Conference on Computer and Information Technology (ICCIT).

[6] Durrani N.,et.al. 2010. Hindi-to-Urdu machine translation through transliteration. In Proceedings of the 48th Annual Conference of the Association for Computational Linguistics, Uppsala, Sweden.

[7] Durrani N, et.al. 2013. Can Markov Models Over Minimal Translation Units Help Phrase-Based SMT? In Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, Sofia, Bulgaria, August. Association for Computational Linguistics.

[8] Durrani N., and Hussain S,. Urdu Word Segmentation. In Proceedings of the 11th Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL HLT 2010), Los Angeles, US (2010).

[9] Islam. Z., 2010 Jörg Tiedemann & Andreas here Eisele: English to Bangla phrase-based machine translation. EAMT 2010: Proceedings of the 14th Annual conference of the European Association for Machine Translation, 27-28 May 2010, Saint-Raphaël, France. Proceedings ed.Viggo Hansen and François Yvon; 8pp.

[10] Hasler,E et.al,. 2012. Sparse Lexicalised features and Topic Adaptation for SMT. in Proceedings of the seventh International Workshop on Spoken Language Translation (IWSLT), pages 268–275.

[11] Heafield. K., 2011. KenLM: Faster and Smaller Language Model Queries. In Proceedings of the Sixth Workshop on Statistical MachineTranslation, pages 187–197, Edinburgh, Scotland, United Kingdom .

[12] Khan, N, et al, English to Urdu Hierarchical Phrase based SMT system.IJCNLP 2013.

[13] Koehn, P,.et al,. Statistical Phrase-BasedTranslation.In HLT-NAACL 2003: conference combining Human LanguageTechnology conference series and the North American Chapter of theAssociation for Computational Linguistics conference series,pages 48–54, Edmonton, AB, 2003.

[14] Koehn, P et.al,. 2007. Moses: Open Source Toolkit for Statistical Machine Translation, Annual Meeting of the Association for Computational Linguistics (ACL).

[15] Moore. R.C. 2002. Fast and accurate sentence alignment of bilingual corpora. In Conference of the Association for MachineTranslation in the Americas (AMTA).

[16] Och, F. J.. 2003. A systematic comparison of various statistical alignment models. Computational Linguistics, 29(1):19–51.

[17] Roy M., 2009. A Semi-supervised Approach to Bengali-English PhraseBased Statistical Machine Translation, Proceedings of the 22ndCanadian Conference on Artificial Intelligence.

[18] Sharma N., Parteek Bhatia, Varinderpal Singh, "English to Hindi Statistical Machine Translation", International Journal in Computer Networks and Security (IJCNS).

[19] Stolcke.A,. 2002. SRILM - an extensible language modeling toolkit. In Intl. Conf. Spoken Language Processing, Denver, Colorado.

[20] Durrani. N, 2006. System for Grammatical Relations in Urdu. Nepalese linguistics, 22:91, 2006.

[21] Trever Cohn and Mirella Lapata. 2007. Machine Translation by Triangulation: Making Effective use of Multi-Parallel Corpora. In the 45th Annual Meeting of the Association for Computational Linguistics, Parague, Czech, June

[22] Bertoldi, N., Barbaiani, M., Federico, M., & Cattoni, R. (2008) Phrasebased statistical machine translation with pivot languages. In International Workshop on Spoken Language Translation Evaluation Campaign on Spoken Language Translation (IWSLT), pp. 143–149.

[23] Nakov, P. and Tiedemann, J. (2012). Combining word-level and character-level models for machine translation between closely related languages. In Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 301–305, Jeju Island, Korea. Association for Computational Linguistics.

[24] Durrani N, Hassan Sajjad, Hieu Hoang, and Philipp Koehn. 2014. Integrating an Unsupervised Transliteration Model into Statistical Machine Translation. In Proceedings of the 15th Conference of the European Chapter of the ACL (EACL 2014), Gothenburg, Sweden April. Association for Computational Ling