HANDLING UNKNOWN WORDS IN URDU TO PUNJABI MACHINE TRANSLATION

Umrinderpal Singh

Department of Computer Science

Punjabi University, Patiala

umrinderpal@gmail.com

Abstract

A Machine translation process automatically converts one language to another language. During the translation process system failed to translate many source language words which are new to the system. Efforts have been made to develop various sub-modules which can help the machine translation system to handle out of vocabulary words and find out a most likely translation in target language. These methods are an enhancement to the baseline Urdu to Punjabi machine translation system. Six different modules have been developed based on various linguistic and morphological rules. Urdu and Punjabi languages are closely related and resource-poor languages. By using methods like segmentation, stemming, creating inflection, handling missing diacritical marks, handling izafaats and transliteration of Urdu words to Punjabi words, increased the overall BLEU score from 0.836 to 0.876 compared to the baseline system.

Key Words: Machine Translation, Rule based approach, Urdu, Punjabi

1. Introduction

The machine translation system translates source language to target language. In this digital era, different communities from across the world interacting with each other and sharing the vast amount of data. All different communities have their local languages and have plenty of knowledge resources in their languages; due to language barriers these different communities are not able to share any language-related resources. In this scenario, machine translation application plays vital role and provide a way to understand each other's language resources and knowledge. The baseline system [Umrinderpal.et.al. 2016] already has been developed for Urdu to Punjabi machine translation using a statistical approach. A large amount of training data is required to train any statistical model, but it is very difficult to include every example in training data and training process also suffers when language pair is part of the resource-poor languages categories like many South-Asian languages. Along with the lack of resources like parallel sentences to train the statistical model. Urdu text also has many different issues like Segmentation, non-standardization of spelling etc. The baseline system treats all these words as out of vocabulary words. When system encountered all these unknown words during translation process it just failed to handle them and return source language words without translation. Therefore, various rule-based modules have been developed to handle these unknown words and try to find possible translations.

2. Methodology

An extension has been developed to the baseline system which contains various sub-modules to handle unknown words during the translation process. When baseline system failed to find any translation in phrase table it submits out of vocabulary word to the unknown word handler module. Unknown word handler module check new word based on various predefined conditions to resolve it. This module consists of different sub-modules and contains various rules to handle unknown words and try to find a valid translation. The baseline Urdu to Punjabi machine translation system had been developed using a statistical approach. When system does not find any translation for source language word in phrase table it returns original word as output. Urdu and Punjabi languages are closely related languages [Umrinderpal et.at 2016,] and share same grammatical structure, vocabulary and morphology. The system tries to find target language translation by analysis of morphological features and by applying various morphological rules of both source and target languages. Efforts have been made to develop six different models to handle these unknown or out of vocabulary words shown in Figure 1. Urdu and Punjabi languages pair is new to the machine transaction system and very less work has been done compared to European languages.

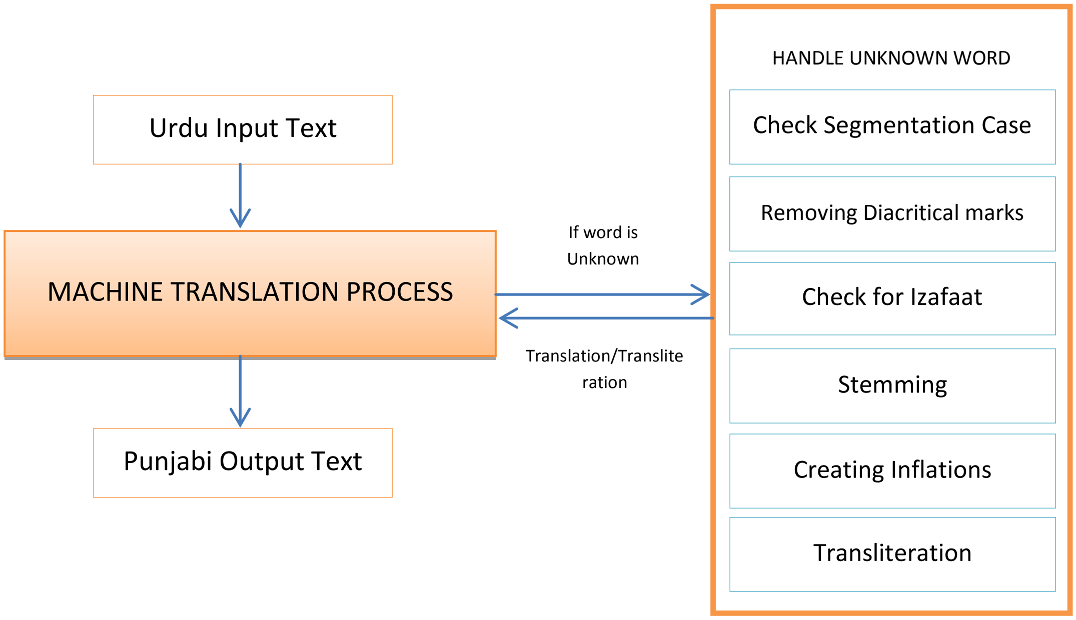

Figure 1: Sub-modules to handle unknown word

Urdu and Punjabi languages are resource poor languages and do not have a significant amount of data to train any statistical model. Therefore, the rules-based approach can be a promising approach to increase the overall accuracy of statistical systems. Rule based system can be combined with baseline system to refine the output in the translation process. Little work has been done by few researchers to develop individual modules for example (Durrani, Nadir and Sarmad Hussain 2010) proposed segmentation method for Urdu and (Lehal, G. 2009 and 2010) proposed Urdu segmentation method for space omission and insertion. (UmrinderPal et.al 2012) proposed various rules to handle named entity for Urdu text.(Rohit Kansal et.al 2012) presented various rules for stemming Urdu word to root form. (Rajeev Puri, Bedi, V.Goyal, 2015) developed a Punjabi stemmer using WordNet databases. A Shahmukhi to Gurmukhi transliteration system has been developed with a significant accuracy which was based on various linguistic rules and databases by (Tejinder Singh Saini and Gurpreet Singh Lehal 2008).

2.1. Checking Segmentation case

Handling segmentation issue is a fundamental and primary challenge in Urdu text processing. In Urdu, there are two types of characters, Joiner, and non-joiners. Where joiner can change their basic space when attached to another character but non-joiner retains its original space (Lehal 2009, 2010) (Durrani, Nadir 2010). Generally, words are separated from each other by whitespace, but in Urdu, writing whitespace becomes optional when a word ends with non-joiner characters. Urdu reader can easily understand this type of writing style but it is difficult for text processing algorithms. The tokenization process is a preprocessing task of any NLP application. Where word tokenization process identifies individual words based on whitespaces, but this word tokenization task become challenging when whitespace does not exist. We have handled few basic types of segmentation issues which have frequently occurred in Urdu text. For example, non-segmented word starts with these prefixes (سے,[se])(کے,[ke])(اور,[aur]) all three ends with non-joiner characters. Sometimes numeric word also acts like non-joiners, and written without any whitespace for example (‘4دن’[four days]) or (1111] 2015مئی may2015]). We have developed various rules to identify word boundaries. In place of finding target language translation, segmentation processing module finds the valid segments of unknown tokens and update original input word list. This word list then processed by translation process and chooses most probable translation based on language and translation model.

ALGORITHM 1. Handling Segmentations issues

```
Algorithm's input
CurrentWord
InputWordList[]
Prefixs[]

FOREACH: prefix in Prefixes[]
    IF: CurrentWord StartWith prefix
        Separate Prefix and Suffix Part
        IF: Suffix Present in PhraseTable
            Remove CurrentWord from InputList
            Update InputWordList[] With New Prefix and Suffix
```

```
                    RETURN True
            ENDIF
    ENDIF
ENDFOREACH

IF: CurrentWord is Alphanumeric
    Apply Rules to separate numeric and Urdu word
    Remove CurrentWord from InputWordList
    Update InputWordList[] With new Tokens
    RETURN True
ENDIF

RETURN NULL  //otherwise
```

2.2. Removing diacritical marks:

In Urdu, diacritical marks are rarely or partially used to write Urdu text (Malik 2006). There are different diacritical marks in Urdu for example Zer, Pesh, Zabar, hamza, Shad, Khari-Zabar, do-Zabar and do-Zer etc. The absence of diacritical marks makes the Urdu text processing a challenging task where one word can yield different meaning for example word ‘اس’ be interpreted ‘this’ as well as ‘that’ and word ‘گل’ can be interpreted as ‘talk’ or ‘flower’ . As mentioned previously diacritical marks are optional in Urdu while writing. Due to this, one word can be write in multiple ways for example word Urdu(اردو) can be written into three forms اُردو ، اُردُو ، اردو . Therefore, one needs to include all variations in training data, but it is not possible to include all the possible variation of all existing words in training examples. If baseline system failed to find out any word containing diacritic marks, then this module removes diacritical marks from unknown word and again tries to find all possible translation in phrase table.

ALGORITHM 2. Handling diacritical marks

```
Algorithm's  input
CurrectWord

DiacriticalMarks[]
InputWordList[]

FOREACH: mark in DiacriticalMarks[]
    IF:  Mark in CurrentWord
    CurrectWord=Remove mark from CurrentWord
    ENDIF
ENDFOREACH

IF: CurrentWord present in PhraseTable
    Update InputWordList[] with CurrentWord
ELSEIF
    RETURN NULL
```

ENDIF

2.3. Checking for Izafaats:

Izafaats are frequently used in Urdu as well as popular in the Punjabi language along with other South-Asian languages. Izafaats can be used to represent proper nouns, designations, etc. For example:

Table 1: Izafats used as Proper Nous

Izafaat	**English Meaning**
وزیر اعظم	**The Prime Minister**
صدر مملکت	**The President**
وراست خالصہ	**Verast-e-Khalsa**

Mostly izafaats can be of two types one which has 'zer'(e sound) and second type contain 'vao' (o sound) character.

Table 2: Izafats having 'zer' and 'vao' sounds

Izafaat	**Romanization**
ضربِ عضب	**zrb-e-ajb**
اعلانِ جنگ	**elan-e-jang**
اتفاقِ رائے	**atfaq-e-raye**
صورتِ حال	**surat-e-haal**
وزارتِ داخلہ	**wazarat-e-dakhla**

غور و فکر	**ghhor- o- fikar**
اب و ہوا	**ab- o- howa**

Izafaats yield meaning by combining two words or bigrams. These bigrams can be attached to each other without any whitespace or it may or may not contain 'zer' character at the end of the first word for example (atfaq-e-raye)اتفاق رائے, (surat-e-haal)صورت حال. The system checked all these conditions and tried to find translation if current bigram is a candidate of an izafaat category. Izafaats may generate reordered translation in Punjabi. For example, اعلانِ جنگ (elan-e-jang) can be translated as **ਲੜਾਈ ਦਾ ਇਲਾਨ**(ldhae da aelan). To handle the Izafaats words, system simply add and removes diacritical marks. For example, if bigram is not present in phrase table, then system creates two versions of current bigrams by adding 'zer' and 'vao' characters in end of the first word of bigrams and again tries to find in phrase table. If current bigram having 'zer' or 'vao' characters in end of the first word but not part of the phrase table then system simply drop these diacritical marks and again try to find in phrase table. If system failed to find translation in both conditions, and bigram contain 'zer' or 'vao' characters then system treat it as proper noun and send it to the transliteration modules.

ALGORITHM 3. Handling Izaafts

```
Algorithm's Input
CurrectWord
CurrecntPosition
InputWordList[]

NextWord=Get Next Word  from InputWordList[++CurrentPostion ]
BiGram=CurrentWord+" "+NextWord
IF: CurrentWord end with [pesh,vow]
        IF: BiGram is present in PhraseTable
                Remove NextWord From InputWordList[] at Postion ++CurrectPosition
                Update Currect word Postion with BiGram
                RETURN true
        ELSE
                Remove NextWord From InputWordList[] at Postion ++CurrectPosition
                Transliteration=GetTransalitration(Bigram)
                Update Currect word Postion with Transliteration
                RETURN true
ELSE IF:  CurrentWord NOT END With [pesh,vow]
      Remove whitespace from BiGram
```

```
        IF: BiGram is Present in PhraseTable
Remove NextWord From InputWordList[] at Postion ++CurrectPosition
                Update Currect word Postion with BiGram
                RETURN true
              ENDIF
RETURN NULL otherwise
```

2.4. Stemming

Stemming is a process of finding root word by truncating the suffix or prefix of a word. Like other Indo-Aryan languages, Urdu and Punjabi are morphological rich language. In Urdu, a word can be inflected in various ways. Therefore, when we deal with morphologically rich languages, we always try to include all examples of inflections in the training data. But it is not always possible to include every inflected form of all words. For example word کتاب(ketab[book]) can be inflected in two-way (ketabe)کتابے, (ketabo)کتبو and word کمرہ(kama[room]) can become [kamare]کمرے, [kamaro]کمرو etc. few more example of inflection and root words shown in the table 3. The system used seven forms of inflections those are frequently employed in Urdu (Rohit Kansal 2012). Rules 1 to 7 have been developed for stemming.

Table 3: Inflections in Urdu

Root Word	Inflection form	English Translation
کتاب(kitaab)	کتابے , کتبو	Book
کمرہ(kamrah)	کمرے , کمرو	Room
لڑکی(larki)	لڑکیاں , لڑکیا	Girl
بستی(bastii)	بستیاں , بستیو	Township
گاڑی(gaari)	گاڑیاں , گاڑیو	Vehicle
میلا(mela)	میلے , میلیو	Fair

Rule 1- If word ends with وں (vao+noon-gunna) then remove وں (vao+noon-gunna) from end.

For example- رنگوں - رنگ
(raṅgōṃ) (raṅg)

Rule 2- If word ends with ے (badi-ye) then remove ے (badi-ye) from end and replace with ا (alif) .

For example- میلے - میلا
(mēlē) (mēlā)

Rule 3- If word ends with یوں (choti-ye +vao+noon-gunna) then remove یوں (chotiye+ vao+noon-gunna) from end and replace with ی (choti-ye).

For example- کویوں - کوی
(kaviyōṃ) (kavī)

Rule 4- If word ends with ؤں (vao- hamza+noon-gunna) then remove ؤں (vao- hamza+noon gunna) from end.

For example- چاچاؤں - چاچا
(cācāōṃ) (cācā)

Rule 5- If word ends with یاں (choti- ye+alif+noon-gunna) then remove یاں (chotiye + alif+noon-gunna) from end and replace with ی (choti-ye).

For example- کوٹیاں - کوٹی
(kōṭīyāṃ) (kōṭī)

Rule 6- If word ends with یں (choti- ye+noon-gunna) then remove یں (choti-ye + noon-gunna) from end.

For example- ڈھالیں - ڈھال
(ḍhālēṃ) (ḍhāl)

Rule 7- If word ends with ئیں (hamza + choti-ye+noon-gunna) then remove ئیں (hamza+chotiye+ noon-gunna) from end.

For example- مالائیں - مالا
(mālāēṃ) (mālā)

The algorithm finds root word translations in phrase table and returns to the decoding module. For example, if input word کمرے (kamre) is not found in phrase table then system generate its root form کمرہ and tries to find a valid translation in target language. Using root form of Urdu word and finding Punjabi translation is not accurate way but it helps one to understand overall meaning of input sentence with a small grammatical mistake. Following algorithm has developed to get root word and its translation.

ALGORITHM 4. Stemming process

```
Algorithm input
CurrentWord
InputWordList[]
SufffixList[]
PrefixList[]
IF:  CurrectWord's Suffix in SuffixList[]
     UrduRootWord=Apply Rules to get root word
     IF:UrduRootWord Present in PhraseTable
     PunjTrans = Get Punjabi Translation from PhraseTable
          PunjabiWord = Apply Target Language Rules to inflect PunjTrans
          RETURN PunjabiWord
                    ENDIF
               ELSE IF: if Currect Word Suffix in PrefixList[]
     UrduRootWord=Apply Rules to get root word
     IF: UrduRootWord Present in PhraseTable
     PunjTrans = Get Punjabi Translation from PhraseTable
          PunjabiWord = Apply Target Language Rules to inflect PunjTrans
          RETURN PunjabiWord
                    ENDIF

               ELSE
                RETURN NULL
```

2.5. Creating inflections of word:

In this module system tries to create all possible inflections of given Urdu word. This module execute after stemming module. When stemming process failed to find a translation of Urdu word. In place of finding root word using stemming, this module does the opposite of stemming by creating inflections of Urdu word by applying various rules (Rohit Kansal 2012). The algorithm treats input word as root form of the word. Algorithm generates all possible inflection of input word and tries to find a valid word in phrase table and return translations. For example, if word کتاب (ketab [book]) is not present in phrase table, then the system generates different forms like کتابے (ketabe) and کتبو (ketabo) then try to find a valid translation.

ALGORITHM 5. Creating inflections

```
  Algorithm's Input
CurrecntWord
```

```
SuffixList[]
PunjabiRoot=NULLl

AllPossibleInflections[] = Generate all Inflection using SuffixList[] and Rules
FOREACH: inflection of AllPossibleInflection[]
        IF: inflection Found in PhraseTable
                PunjabiWord=Get Punjabi Translation from PhraseTable
        PunjabiRoot=Apply Target language Rules to get root word of PunjabiWord.
        ENDIF
ENDFOREACH

            RETURN PunjabiRoot
```

2.6. Transliteration

Transliteration process has been used to change the Urdu script to Gurumukhi script (Saini and Lehal 2008, 2010). Urdu used Arabic script for writing and Punjabi used Gurumukhi script. The System used various mapping rules to map Urdu characters to Punjabi characters. This module consists of 353 mapping rules. Rules contained one-to-one, many-to-one, one-to-many, many-to-many mapping rules. In mapping rules, five-grams were the maximum Urdu language characters length which was mapped with Punjabi's Gurmukhi characters. All rules have been developed manually by analysis of Urdu and Punjabi text. Transliteration process applied rules on an unknown word in decreasing order of length from five-gram to one-gram.

A few rules were made according to target language inflection, where the system had mapped Urdu word inflection to Punjabi word infections which yielded target language word close to translation. For example: if Urdu word ends with ایا that would be replaced with ◌ਾਇਆ, if Urdu word ends with وں then replace it with ◌ੂਆਂ. A few suffix mapping rules are shown in table 4. This module helps in converting proper names to Punjabi script along with foreign word which is common in both languages. Transliteration module is the last step to handle the unknown word, therefore if Urdu word is new to the system and every module failed to identify Urdu word, then this model simply returns transliteration form of the Urdu word in Punjabi script.

Table 4: Urdu Punjabi Suffix Mapping

S.No	Urdu suffix	Punjabi suffix
1	وں	◌ਾ◌ਂ
2	یوں	ਅ◌ਾ◌ਂ
3	ؤں	ਿ◌ਅ◌ਾ◌ਂ
4	یاں	◌ੀਅ◌ਾ◌ਂ
5	یں	◌ਾ◌ਂ

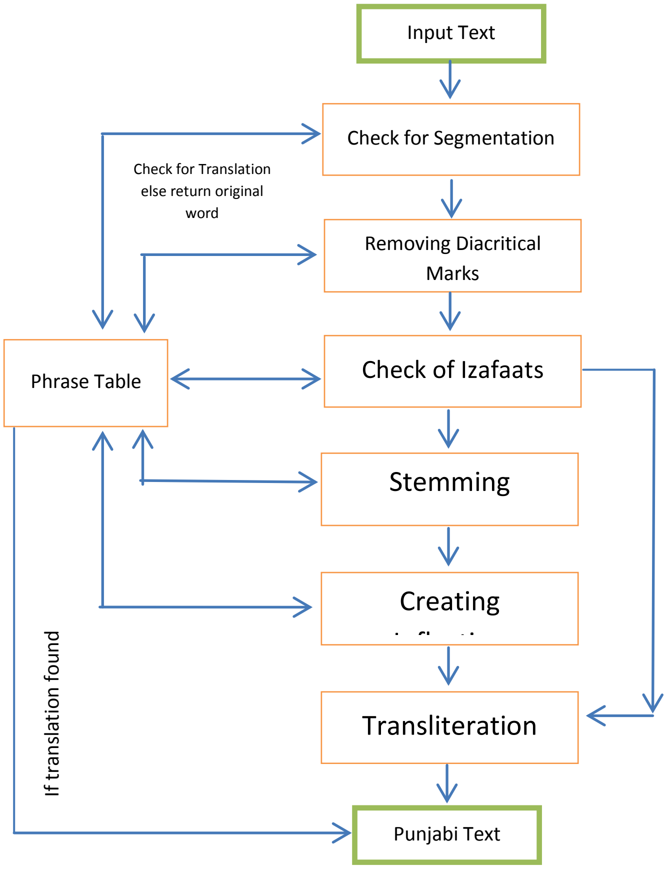

Figure 2: Inter module processes to handle unknown words

3. Evaluation

The system has been evaluated using BLEU score and manual evaluation methods and compared with baseline system. Where BLEU score is automatic evaluation metric score range from 0 to 1 and for manually checking we have set four parameters as shown below in table 5. The system's output was compared for five domains sports, entertainment, health, politics, and tourism. All testing data was collected from BBC Urdu News website. Detail of the testing data shown in table 6.

Table 5: Manually Evaluation Score

Score	Cause
0	Very poor
1	Partially Okay
2	Good with few errors
3	Excellent

Table 6: Testing Data

Domain	News Documents	Sentences
Political	10	792
Sports	10	631
Entertainment	10	690
Tourism	10	708
Health	10	613

For evaluation, test set contained ten documents for each domain. Total 3434 sentences were there in a test set by combining all documents. In manual testing, 87% sentences scored 3 and 2 and 11% sentences scored one. The rest got 0 scores. BLEU score of this test set was 0.876. Results show that handling the unknown words using various rule-based modules can clearly increase the overall accuracy in all domains. The baseline system returned original words in target language output text without any translation which was not part of the training data. The majority of these words were segmentation cases, proper nouns, inflected forms of Urdu words and foreign words from English. The system was able to get 0.876 BLEU score which is better than baseline system. The baseline system got 0.836 BLEU score. The proposed system also failed in different modules, most of the errors occurred during transliteration process where source characters were not mapped to correct target script character. There are various one-to-many mappings between source and target characters and system failed to choose correct transliteration for a particular character.

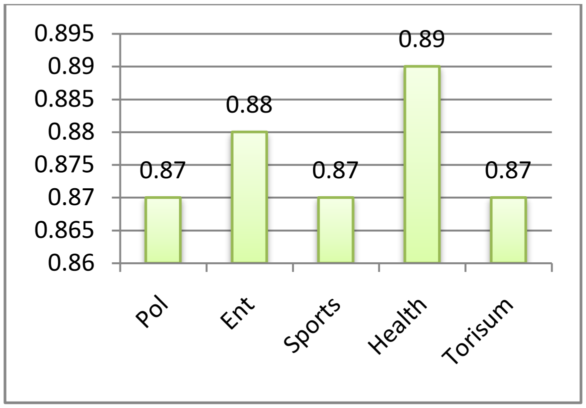

Chart 1: Per domain accuracy

4. Conclusion

This Paper presented six different methods to handle unknown words in Urdu to Punjabi Translation process. Experiments show that in closely related and morphologically rich languages unknown words can be handled using various rules based sub-modules along with main translation system. We have presented a way to handle izafaats, inflected word, segmentation issues, and variations of missing diacritical marks words. Presented methods show increase in overall BLEU score as compared to the baseline system. This extended system managed to get 0.876 BLEU score using all proposed modules. In future work, all these module and rules can be upgraded by adding more rules in all sub-modules especially in segmentation module which is a quite challenging task for Urdu text processing applications. The system includes various rules for transliteration process but this process can be more accurate by adding and updating linguistic rules for Urdu to Punjabi transliteration.

REFERENCES

[1] Durrani, Nadir, and Sarmad Hussain. Urdu word segmentation, Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics. Association for Computational Linguistics. (2010)

[2] Kansal Rohit, Vishal Goyal, G.S lehal, Rule Based Urdu Stemmer, Proceedings of COLING 2012: Demonstration Papers, pages 267-276 2012

[3] Lehal, Gurpreet Singh. A word segmentation system for handling space omission problem in Urdu script." 23rd International Conference on Computational Linguistics (2010).

[4] Lehal, G. S. A Two Stage Word Segmentation System for Handling Space Insertion Problem in Urdu Script. World Academy of Science, Engineering and Technology 60 (2009).

[5] Lehal, G. S. and Saini, T. S. A Hindi to Urdu Transliteration System. In Proceedings of 8th International Conference on Natural Language Processing, pages 235-240, Kharagpur, India. (2010)

[6] Misbah Akram, & Sarmad Hussain.. Word segmentation for urdu OCR system, Proceedings of the 8th Workshop on Asian Language Resources, Beijing, China (2010).

[7] Malik, M. G. A. Punjabi Machine Transliteration. In Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the ACL 1137-1144. (2006)

[8] Malik, A., Boitet, C. and Bhattacharyya, P. Hindi Urdu machine transliteration using finite-state transducers. In Proceedings of the 22nd International Conference on Computational Linguistics, pages 537–544, Manchester, UK. (2008)

[9] Poowarawan, Y.. Dictionary-based Thai Syllable Separation. Proceedings of the Ninth Electronics Engineering Conference (1986).

[10] Puri Rajeev, Bedi, Vishal .Goyal. Punjabi Stemmer using Punjabi WordNet Database, Indian Journal of Science and Technology, Vol 8(27) **(2015)**

[11] Richard Sproat, C. S. A Stochastic FiniteState Word-Segmentation Algorithm for Chinese.Computational Linguistics. (1996)

[12] Sproat, R., Shih, C., Gale, W., & Chang, N. A Stochastic Finite-State Word-Segmentation Algorithm for Chinese. omputational Linguistics. (1996)

[13] Saini, Tejinder Singh and Gurpreet Singh Lehal. Shahmukhi to Gurmukhi Transliteration System: A Corpus based Approach, Advances in Natural Language Processing and Applications Research in Computing Science 33, 2008, pp. 151-162 (2008)

[14] Singh, Umrinderpal, Vishal Goyal and Gurpreet Singh Lehal, Urdu to Punjabi Machine Translation: An Incremental Training Approach, International Journal of Advanced Computer Science and Applications(IJACSA), 7(4). 227-238 (2016)

[15] Singh, Umrinderpal et.al "Named Entity Recognition System for Urdu". Processing of Colling 2012 pages: 2507-2518 (2012)