Urdu to Punjabi Machine Translation: An Incremental Training Approach

Umrinderpal Singh
Department of Computer Science,
Punjabi University
Patiala, Punjab, India

Vishal Goyal
Department of Computer Science,
Punjabi University
Patiala, Punjab, India

Gurpreet Singh Lehal
Department of Computer Science,
Punjabi University
Patiala, Punjab, India

***Abstract*—The statistical machine translation approach is highly popular in automatic translation research area and promising approach to yield good accuracy. Efforts have been made to develop Urdu to Punjabi statistical machine translation system. The system is based on an incremental training approach to train the statistical model. In place of the parallel sentences corpus has manually mapped phrases which were used to train the model. In preprocessing phase, various rules were used for tokenization and segmentation processes. Along with these rules, text classification system was implemented to classify input text to predefined classes and decoder translates given text according to selected domain by the text classifier. The system used Hidden Markov Model(HMM) for the learning process and Viterbi algorithm has been used for decoding. Experiment and evaluation have shown that simple statistical model like HMM yields good accuracy for a closely related language pair like Urdu-Punjabi. The system has achieved 0.86 BLEU score and in manual testing and got more than 85% accuracy.**

Keywords*—Machine Translation; Urdu to Punjabi Machine Translation; NLP; Urdu; Punjabi; Indo-Aryan Languages

I. Introduction

The machine translation is a burning topic in the area of artificial intelligence. In this digital era where across the world different communities are connected to each other and sharing a vast amount of resources. In this kind of digital environment, different natural languages are the main obstacle to communicate. To remove this barrier researcher from different countries and big companies are putting efforts to develop machine transition system to resolve this barrier. Various kinds of approaches have been developed to decode natural languages like Rule based, Example-based, Statistical and various hybrid approaches. Among all these approaches, statistical based approach is a quite dominant and popular in the machine translation research community. The statistical systems yield good accuracy as compared to other approaches but statistical models need a huge amount of training data. In comparison to European languages Asian languages are resources poor languages therefore it is challenging task to collect parallel corpus for training these statistical model. There are many machine translation systems which have been developed for Indo-Aryan languages [Garje G V, 2013]. Most of the work have been done using rule-based or hybrid approaches because the non-availability of resources. The proposed system based on an incremental training process for training the machine learning algorithm. Efforts have been made to develop parallel phrase corpus in place of parallel sentence corpus. Collecting parallel phrases were more convenient as compared to the parallel sentences.

II. Urdu and Punjabi: A Closely Related Language Pair

Urdu[2] is the national language of Pakistan and has official language status in few states of India like New Delhi, Uttar Pradesh, Bihar, Telangana, Jammu and Kashmir where it is widely spoken and well understood throughout in the other states of India like Punjab, Rajasthan, Maharashtra, Jharkhand, Madhya Pradesh and many other[1] . The majority of Urdu speakers belong to India and Pakistan, 70 million native Urdu speakers are in India and around 10 million speakers in Pakistan[2] and thousands of Urdu speakers living in US, UK, Canada, Saudi Arabia and Bangladesh. The word Urdu is derived from Turkic word ordu which means army camp[2]. The Urdu language was developed in 6th to 13th century. Urdu vocabulary mainly derived from Arabic, Persian, and Sanskrit and it is very closely related to modern Hindi language. Urdu is written in Nastaliq style and script is written from right to left using heavily derided alphabets from Persian which is an extension of Arabic alphabets. [3]Punjabi is an Indo-Aryan language and 10th most widely spoken language in the world there are around 102 million native speakers of Punjabi language across worldwide[4]. Punjabi speaking people mainly lived in India's Punjab state and in Pakistan's Punjab. Punjabi is the official language of Indian states like Punjab, Haryana, and Delhi and well understood by many other northern Indian regions. Punjabi is also a popular language in Pakistani Punjab region but still did not get official language status. In India, Punjabi is written in Gurmukhi script means from Guru's mouth and in Pakistan Shahmukhi is used means from the king's mouth. Despite from the different scripts use to write Punjabi, both languages share all other linguistics features from grammar to vocabulary in common.

Urdu and Punjabi are closely related languages and both belong to same family tree and share many linguistic features like grammatical structure and vast amount of vocabulary etc. for example:

Urdu: - وہ پنجابی یونیورسٹی کا طالب علم ہے

Punjabi: ਉਹ ਪੰਜਾਬੀ ਯੂਨੀਵਰਸਿਟੀ ਦਾ ਵਿਦਿਆਰਥੀ ਹੈ ।

English: He is a student of Punjabi University.

Despite from script and writing order where Urdu is written in right to left using Arabic script and Punjabi from left to right using Gurumukhi script, every other linguistic feature is the same in both sentences. Both sentences shares same grammatical order and most of the vocabulary, this is also true in care of more complex sentences. By analysis of both languages, we found that both languages share many similarities and are used by a vast community of India and Pakistan. Therefore, we need a natural language processing system which can help these people to share and understand text and knowledge. The efforts have been made to develop a machine translation system for Urdu to Punjabi text to overcome this language barrier between both the communities. With the help of this machine translation system, native Punjabi reader can understand Urdu text by translating into Punjabi text.

III. Challenges to Develop Urdu to Punjabi MT System

A. ***Resource poor languages:*** Urdu and Punjabi languages are new in natural language processing area like any other Indo-Aryan language. Both languages are resource-poor language, very small or no annotated corpus is available for development of a full-fledged system.

To develop a machine translation system based on the statistical model, one should need a huge parallel corpus to training the model. For rule-based approach or hybrid machine translation system, one should need a good part of speech tagger or stemmer and large parallel dictionaries. To best of our knowledge, Urdu-Punjabi language pair does not have these resources in a vast amount to train or develop the system. Therefore, development of resources is one of the key challenges to work on this language pair.

B. **Spelling variation:** Due to lack of spelling standardization rules, there are many spelling variation for the same word. [Singh, UmrinderPal et.al 2012] Both languages use tons of loan words from English. Therefore, many variations come in existence, for example, word 'Hospital' can be written in two ways in Urdu اسپتال/ ہسپتال hasptaal/asptaal. It is always a challenging task to cover all variation of a word. There is no standardization in spelling. Therefore, it all depends on a writer which spelling he/she choose to write foreign language words.

C. **Free word order:** Urdu and Punjabi are free word order languages. Both languages have unrestricted word order or phrase structures to form the sentences that make the machine translation task more challenging. For example,

Urdu: رام نے ستا کو اپنی کتاب دی

Transliteration: raam ne satta ko apanee kitaab dee.

English: Ram gave his book to Sita.

This can be rewritten as following:

Urdu: رام نے دے دی ستا کو اپنی کتاب

Transliteration: raam ne dee sata ko apanee kitaab.

Urdu: رام نے دے دے اپنی کتاب ستا کو

Transliteration: raam ne de apanee kitaab sata ko.

Urdu: رام نے اپنی کتاب ستا کو دے دی

Transliteration: raam ne apanee kitaab sata de dee.

Above example shows that same sentence can be written in various ways due to free word order and all sentences give exactly the same meaning. Therefore, it is always difficult to form every possible rule to interpreter's source language text to do machine translation.

D. **Segmentation issues in Urdu:** Urdu word segmentation issue is a primary and most significant task [Lehal, G. 2009]. Urdu is effected with two kinds of segmentation issues, space insertion and space omission [Durrani, Nadir et.al. 2010]. Urdu is written in Nastaliq style which makes the white space completely an optional concept. For example,

Non-Segmented: ا قافلےکےصدراحمدشیرڈوگرانےکہ

Segmented Text: قافلے کے صدر احمد شیر ڈوگرا نے کہ

Urdu reader can read this non-segmented text easily but this is still difficult for computer algorithms to understand. In preprocessing phase, modules like tokenization need to identify individual words for further processing, without resolving the segmentation issue, no NLP system can process Urdu text efficiently and yield less accuracy.

E. **Morphological rich languages:** Urdu and Punjabi are morphological rich languages, where one word can be inflected in many ways. For example, word 'chair' کرسی/kursi can take any form like کرسیا/kursiya, کرسیو/kurseo, کرسئے/kurseye etc. One should need to incorporate all the inflation in our knowledge base to translate them into the target language. Adding all the inflation forms of all words in training data is a big challenge otherwise, it will effect on the accuracy of the system.

F. **Word without diacritical marks:** Urdu has derived various diacritical marks from Arabic to produce vowel sounds, like Zabar, Zer, Pesh , Shad , hamza , Khari-Zabar, do-Zabar and do-Zer [Sani, Tajinder Singh 2011]. In naturally written text diacritical marks are used very rarely. Due to missing of diacritical marks, an Urdu word can be mapped to many different target language translations, for example, word dil/دل often used without diacritical marks and can be interpreted as 'Heart' and 'DELL' without knowing the context of this word. Missing of diacritical marks is a key challenge to choose a proper translation in the target language and the system always needs to disambiguate these words. Along with this, the missing diacritical marks create various variations of the same word, for example, word 'Urdu' can be written in three ways(اردو)(اُردو)(اُردُو). Therefore, one should need to include all of these variations in the training examples.

1. https://en.wikipedia.org/wiki/States_of_India_by_Urdu_speakers
2. https://en.wikipedia.org/wiki/Urdu
3. https://en.wikipedia.org/wiki/List_of_languages_by_number_of_native_speakers
4. https://en.wikipedia.org/wiki/Punjabi_language

IV. METHODOLOGY

An Incremental machine learning process has been used, in place of manually developed parallel sentences corpus of source and target languages. Urdu and Punjabi languages are resource-poor language; the non-availability of the parallel corpus is a primary challenge to develop a statistical machine translation system. Efforts have been made to develop a corpus of manually mapped parallel phrases. Figure 1 shows the overall learning process of machine translation systems. The system takes Urdu text document as input and translates using initial uniformed distributed data. Initially, the system has phrase tables for most frequent 5000 Urdu words mapped with Punjabi translations. Due to insufficient data in phrase tables, many Urdu words returned without translation in parallel phrase file generated by decoding module shown in Appendix 1.

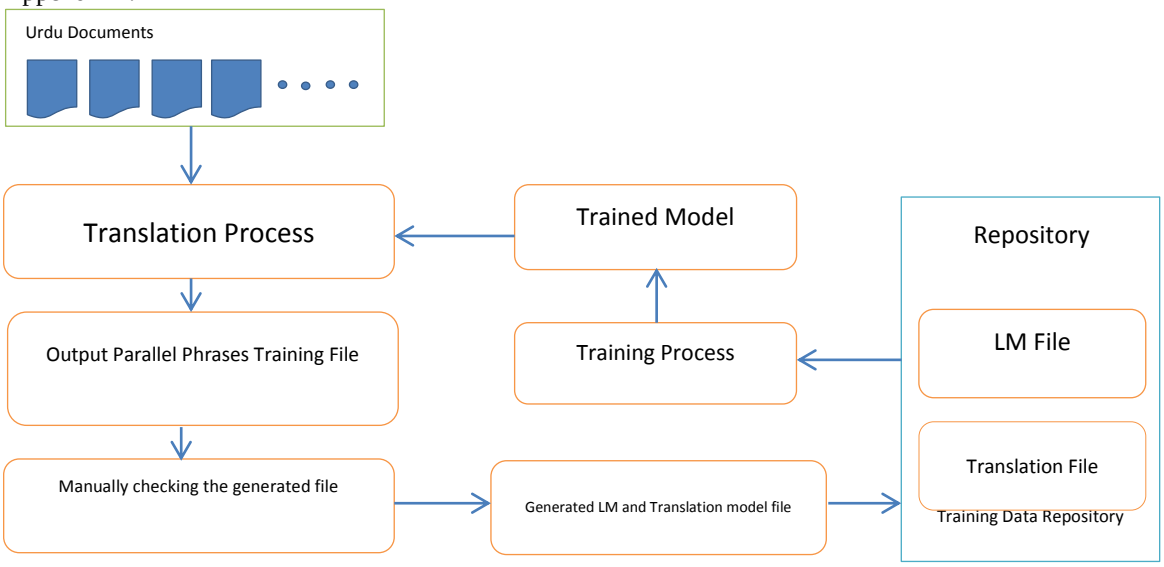

Fig. 1. Incremental MT training and decoding system

Then generated file manually corrected and updated with new translations by linguists. This updated file again submitted to the system to generate language model and translation model. The system learns new parameters from all the updated all files present in the repository of generated training files. Then system updates language model and phrase tables with a new vocabulary and update probabilities. With this incremental learning process, the system gets trained by each document it processes, learn and update language and translation model. The complete system is divided into five different processes or modules, Tokenization and segmentation, Text classification, Translation model learning, language model learning and decoding process.

A. **Tokenization and segmentation process:** Tokenization process is the primary and most significant task of any machine translation system. In preprocessing phase, the input text is divided into isolated tokens or words by tokenization process based on whitespace. Tokenization process is also a challenging task to identify valid tokens, when the system has noisy input data. Where tokens are often attached to neighboring tokens without any whitespace in-between them. This kind of writing trend is quite common in Urdu, where whitespace is an optional thing. The proposed tokenization process works on two levels, (1) isolates sentence boundary identification and (2) isolate word boundary identification.

1) **Tokenization into Sentences:** In sentence tokenization process, the system identifies sentence boundary based on few symbols used in Urdu to complete the sentence. For example, Urdu sentences often end with,{ ؟ ,۔ }, but symbol { ۔ } is an ambiguous one and not always used to identify the sentence boundary. This symbol { ۔ } also used as a separator in abbreviations. For example, آئی۔ سی۔ سی۔ , therefore, to tokenize text into sentences few rules were formed to check boundary conditions based on abbreviation. For example, the system always checks surrounding words of sentence termination symbols in abbreviation list.

2) **Tokenization into words:** The word tokenization process identifies individual tokens or words in the input text. To identify all the individual tokens first, one should need to separate all the words from symbols which are attached to words. For example, the system inserts whitespace in-between symbols and words and change them from آئے ہیں۔ to آئے ہیں ۔ .

ALGORITHM 1. Tokenization and Segmentation Process

```
Read Input Text in InputText
FinalList[]
Sentences[][]

        Insert space between word and symbols
        Tokenization InputText into Partial_Token_list[] form whitespace

        LOOP: Partial_Token_list[]
                IF: Current word is alphanumeric
                        Apply rules to word into split numeric and suffix part.
                        Add word in FinalList[]
                ELSE IF: Current word length > 3 and start with {کے , اور , سے}  and word not present in DB
                        Apply rules to split prefix and suffix parts
                        IF: suffix part is present in Phrase Table
                                Add prefix and suffix words in FinalList[].
                        END IF
                ELSE
                        Add word in FinalList[]
        END LOOP

        LOOP: FinalList[]
                IF: Current token is not a sentence separator
                        Sentence += token+" "

                ELSE IF:  Current token is a sentence separator AND previous and next are not abbreviation tokens
                        Add Sentence in Sentences[][]
        END LOOP
```

3) **Segmentation process:** The segmentation issue is a key challenge in Urdu text processing NLP applications. Segmentation issue can be handled on two levels, space insertion and space omission as discussed in MT challenges. In tokenization process, the system has handled only space insertion issue. Space omission problem is negligible in Unicode Urdu text but space insertion is quite frequent. To resolving the word segmentation problem in Urdu is quite a challenging task and need a full-fledged algorithm for this. Rather than handling all segmentation issues, the system has handled most frequent cases of segmentation. For example, in Urdu text, most of the time word attached with these prefixes {**کے, اور, سے**} which are ends with non-connecters and easily understood by Urdu reader but difficult for a computer algorithm to process. Few examples of segmentation words start with these prefixes are { **کےلیے, کےبعد, اورترک , اورنام , سےپہلے , سےکہیں** }. The analysis shows that these three words were 65% of all segmentation cases found in Urdu text and 5% cases of segmentation were related to alphanumeric words. Alphanumeric segmentation issue is also quite common in Urdu text, for example,{ **دسمبر21سے**،26}. Various rules have been developed to handle these types of tokens.

B. **Text Classification:** Most of the statistical machine translation system use single phrase table for translation. Instead of single phrase table for translation, the proposed system has used five different phrase tables for each domain. The system has trained on political, health, entertainment, tourism and sports domains. After tokenization process, text classifier needs to classify input text into most probable class, then translation module uses specific domain phrase table to translate input text. The text classifier returns a list of all domains with the higher probable domain on top followed by less probable domains. Other domains are used as a backoff model when the system did not find an Urdu phrase in the top domain then it searches in next less probable domain and so on.

$$C(punjabi\ phrases) = \begin{cases} phrase\ translation\ \ if\ domain = x1 \\ phrase\ translation\ if\ \ domain = x2 \\ phrase\ translation\ \ if\ domain = x3 \\ phrase\ translation\ if\ \ domain = x4 \\ phrase\ translation\ \ if\ domain = x5 \\ \quad else\ return\ original\ phrase \end{cases}$$

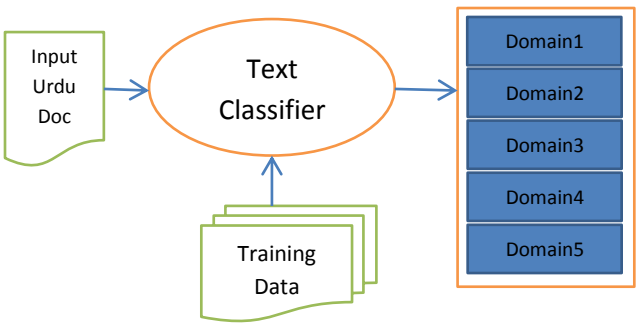

Fig. 2. Text classification system

Naïve Bayes model has been used to classify the input text, Naïve Bayes model considers document as bag of word where word positions are not important for classification, The Naïve Bayes approach based on Bayes rule defined as:

$$C = argmax_{c \in C} \frac{P(d|c)P(c)}{P(d)} \quad (1)$$

Rewriting by dropping the denominator because of constant factor:

$$= argmax_{c \in C} P(d|c)P(c) \quad (2)$$

To representing features of the documents for a class, equation can be written as:

$$= argmax_{c \in C} P(x_1, x_2, \dots, x_n|c)P(c) \quad (3)$$

Joint probability of whole set of independent features defined as:

$$P(x_1, x_2, \dots, x_n|c) = P(x_1|c) * P(x_2|c) * P(x_3|c) * \dots * P(x_n|c) \quad (4)$$

Simplified as:

$$C_{NB} = argmax_{c \in C} P(c_j) \prod_{x \in X} P(x|c) \quad (5)$$

To calculate maximum likelihood estimate and prior defined as:

$$P(w_i|c_j) = \frac{count(w_j, c_j)}{\sum_{w \in V} count(w, c_j)} \quad (6)$$

$$P(c_j) = \frac{Doc_{count}(C = C_j)}{N_{doc}}$$

To handle the unknown words, classifier has used Laplace smoothing defined as:

$$P(w_j|c) = \frac{Count(w_i, c) + \lambda}{\sum_{w \in V} count(w, c) + \lambda} \quad (8)$$

Rewritten as:

$$P(w_j|c) = \frac{Count(w_i, c) + \lambda}{\sum_{w \in V} count(w, c) + |V|} \quad (9)$$

Where $|V|$ is size of vocabulary and λ is constant value to add in frequency count of word in a document.

The system has used a list of 100 stop words to remove uninformative words which are common in training examples. Urdu is a morphologically rich language and one word can appear in the corpus with different suffixes, therefore, to transform all inflected words to root form in the training examples Urdu stemming rules has been used [Rohit Kansal et.al 2012].

C. **Translation and Language model Training:** The machine translation system's training process is divided into two main parts, Translation model, and Language model learning. The system used Hidden Markov Model (HMM) as learning process and Viterbi algorithm as a decoder.

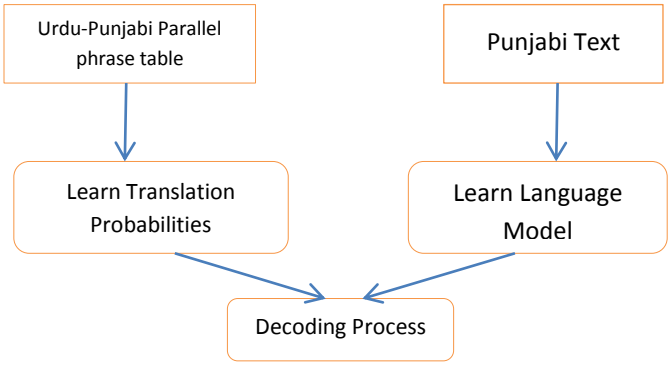

Fig. 3. Statistical machine translation model

HMM is a generative model defined as:

$$f(s_1 \dots s_n) = \underset{t \dots t_n}{\text{argmax}} P(s_1 \dots s_n, t_1 \dots t_n) \quad (10)$$

Where $s_1 \dots s_n$ are source language phrases and $t_1 \dots t_n$ target language phrases. By inputting the $s_1 \dots s_n$, we take the highest probability phrase sequence as output of target language. One should define bigram HMM model as below:

$$p(s_1 \dots s_n, t_1 \dots t_n) = \prod_i q(t_n|t_{n-1}) \prod_i e(s_i|t_i) \quad (11)$$

$$q(t_n|t_{n-1}) = \frac{Freq(t_{n-1}t_n)}{Freq(t_{n-1})} \quad (12)$$

$$e(s_i|t_i) = \frac{Freq(t_i \rightarrow s_i)}{Freq(t_i)} \quad (13)$$

1) **Translation model:** Urdu and Punjabi languages are closely related languages. Both languages share identical grammatical structure as well as same word order [Durrani, Nadir et.al 2010]. To learn the translation model we have manually mapped the phrases of source and target languages. Where IBM models provide an elegant solution to automatically mapped source and target language phrases, but for that, one should really need a large parallel corpus to train the model. Urdu and Punjabi are resource poor languages as we discussed in challenges. Therefore, the efforts have been made to find out a simple and effective solution for the training process.

The system takes manually mapped phrases as a training file and calculates translation probabilities. Sample of a training file is shown in appendix 1.

For example: word اتفاق can translate into four different ways.

TABLE I. POSSIBLE TRANSLATIONS

Urdu Word	Punjabi Word
اتفاق	ਸਹਿਮਤ
	ਸਹਿਮਤੀ
	ਸਹਿਯੋਗ
	ਹਮਾਇਤ

Maximum likelihood estimation of word اتفاق .

$$P_{Urdu}(punj)\begin{cases}0.19047619048 \ if\ punj = \text{ਸਹਿਮਤ}\\ 0.17460317460 \ if\ punj = \text{ਸਹਿਮਤੀ}\\ 0.49206349206 \ if\ punj = \text{ਸਹਿਯੋਗ}\\ 0.14285714286 \ if\ punj = \text{ਹਮਾਇਤ}\end{cases}$$

$$P(phrase) = \sum_i P_{Urdu}(punj)_i = 1 \quad (14)$$

TABLE II. POSSIBLE TRANSLATION WITH PROBABILITY VALUES

Urdu Words	P(punj|urdu)
اس	ਇਸ (0.53138492195)
	ਇਹ (0.4251 793756)
	ਉਸ (0.04350714049)
سفر	ਸਫਰ(1.0)
میں	ਮੈਂ (0.0193076817)
	ਵਿਚ (0.0013791201)
	ਵਿੱਚ (0.98055440629)
وہ	ਉਹ (1.0)
پہلا	ਪਹਿਲਾ (1.0)
میچ	ਮੈਚ (1.0)

$$p(p|u) = q(\text{ਇਸ}|\text{اس}) * q(\text{ਸਫਰ}|\text{سفر}) * q(\text{ਵਿੱਚ}|\text{میں}) * q(\text{ਉਹ}|\text{وہ}) * q(\text{ਪਹਿਲਾ}|\text{پہلا}) * q(\text{ਮੈਚ}|\text{میچ})$$

= 0.53138492195*1.0*0.98055440629*1.0*1.0*1.0

= 0.521051826

If training algorithm knows mapping in advance then it is quite straightforward to calculate translation probabilities from their occurrence in training data. In proposed method, the training algorithm already has alignments of all phrases, therefore; it can calculate parameters for the generative model.

$$P(phrase_i) = \frac{Count(phrase_i)}{\sum Count(*)} \quad (15)$$

Appendix 1 shows one-to-one, one-to-many, many-to-one, many-to-many word mapped phrases. In training data, we try to combine multiple words into a phrase which are frequent or combined words yield valid translation in target language. To compare with IBM models, we have used 50000 thousand parallel Urdu-Punjabi sentences to train the model using Moses toolkit which used Giza++ for phrase alignment. For 50000 sentences Moses generated over 3168873 phrases of size 503 MB. By examined generated phrase table manually and found many miss alignments and unnecessary long phrases those were increasing the size of phrase table and adding complexity to search space for decoding algorithm. As compared to an automatically generated phrase table, our manually mapped phrase table for the same set of sentences contains 56023 thousand phrases which are sufficient to translate given sentences accurately of that domain as shown in experiment section. In our phrase table, a maximum length of any phrase was four-gram and total four-gram phrases was 1093 compared to automatically generated phrase table contain several thousands of four-gram phrases.

Automatically find the alignment of words and phrases using parallel corpus is a graceful solution but when we deal with resource-poor languages we need to find out alternative ways. Development of machine learning resources like sentence-aligned parallel corpus is a time-consuming job. To train any machine translation model; one should require millions of parallel sentences. Therefore, if one do not have parallel corpus it is better idea to map phrases rather than writing parallel sentences. Mapping and checking phrases incrementally makes the job easier. Mapping the phrases gave you three advantages first you just need to write a short phrase in place of the whole sentence in the target language. During training processes system generate partial translation or nearly complete translation of an input document. We just need to check or mapping new words in generated files. Second is your phrase table size will be very small compared to automatically generated phrase table it will make a decoding process more efficient. Third, a linguistic person needs less time to generate parallel phrases then parallel sentences.

2) **Language model:** The language model is responsible for generating natural language. The system has been used Kneser-Ney smoothing algorithm to generate language model (Chen and Goodman 1998). Kneser-Ney is an extension of Absolute Discounting and provides state of the art solution for predicting next word. Absolute Discounting method is defined as:

$$P_{AbsDis}(w_i|w_{i-1}) = \frac{c(w_{i-1}, w_i) - d}{c(w_{i-1})} + \lambda(w_{i-1})P(w) \quad (16)$$

Kneser-Ney is a refined version of Absolute Discounting and gave a better prediction on lower order models when higher order modes have no count present. Following equation shows the second order Kneser-Ney model.

$$P_{KneserNey}(w_i|w_{i-1}) = \frac{\max(c(w_{i-1}, w_i) - d, 0)}{c(w_{i-1})} + \lambda(w_{i-1})P_{Conti}(w_i) \quad (17)$$

Where λ is normalized constant, defined as:

$$\lambda w_{i-1} = \frac{d}{c(w_{i-1})}|\{w: c(w_{i-1}, w) > 0\}| \quad (18)$$

Where $\{w: c(w_{i-1}, w) > 0\}$ is number of word types that can fallow, w_{i-1}.

$P_{Conti}(w_i)$ used as a replacement of maximum likelihood of unigram probabilities with continuation probability that estimate how likely the unigram is to continue in a new context. Continuation probability distribution defined as:

$$P_{Conti}(w) = \frac{|\{w: c(w_{i-1}, w) > 0\}|}{\sum_{w'}|\{w': c(w'_{i-1}, w') > 0\}|} \quad (19)$$

$|\{w: c(w_{i-1}, w) > 0\}|$: Where numerator equation is a count of different word types before the word w.

$\sum_{w'}|\{w': c(w'_{i-1}, w') > 0\}|$: Denominator equation is a normalized factor, total count of different words preceding the all words. Recursive formation of kneser-Ney for higher order model defined as:

$$P_{KneserNey}(w_i|w_{i-n+1}^{i-1}) = \frac{\max\left(C\left(w_{i-n+1}^{i-1}\right) - d, 0\right)}{C\left(w_{i-n+1}^{i-1}\right)} + \lambda(w_{i-n+1}^{i-1}) P_{Conti}(w_i|w_{i-n+1}^{i-1}) \quad (20)$$

To form the language model we have used a mixture of phrase and word-based language model. Generally, machine translation systems and other NLP applications used word-based language model. We have tried to develop phrase-based model along with word-based model which gives accurate predictions to choose correct phrases or word to generate target language. The system generates phrase separator training data files to generate phrase and word-based language model file shown in Appendix 2. Changes have been made in language model training data to reduce vocabulary size. For example, we have changed all numeric tokens with a unique token like 22.201 and 545.1 numeric values with 11.111 and 111.1 respectively. Changing the numeric token with unique tokens helped smoothing algorithm to efficiently predict phrase sequence with the same pattern with different numeric tokens for example.

He paid $50 to shopkeeper.

He paid $30 to shopkeeper.

Both these sentences changed to:

He paid $11 to shopkeeper.

Along with numeric patterns, we changed patterns like an email address to unique token [e@e] which helped us to decrease the size of a language model.

D. **Decoding:** Decoding problem find the most likely state sequence from given observation $O = o_1, o_2, o_3 \ldots . o_n$, to decoding the Hidden Markov Model and find the state sequence with the maximum likelihood the system had used Viterbi algorithm. The sequence of states is backtracked after decoding the whole sequences.

ALGORITHM 2. Viterbi

Input: a Sentence
$x_1 \ldots x_n$ *and Parameters* $q(t_n|t_{n-1}), e(s_i|t_i)$
Define K to set of all tags. $K_{-1=}K_0 = (start)$
$\pi(0, start, start)$=1
For $k = 1 \ldots . n$
For $a \in K_{k-1}, b \in K_k$
$\pi(k, a, b) = argmax(\pi(k - 1, a, b) * q(b|a) * e(x_k|b))$
Return $\arg\max(\pi(n, b) * q(stop|b))$

ALGORITHM 3. Complete Translation Process

Read input in UrduInputText
Tokenization and Segmentation UrduInputText in TokensList[]
Classify TokensList[] Text in Classes[]
Load DomainPhraseTables[] according to Classes[]
Load LanguageModel[]

For each Token in Tokens[]
Decode TranslationModel[] and LanguageModel[] using Veterbi

End For
Return Translation

V. Experiment and Evaluation

The system has been evaluated using BLEU score which is automatic evaluation metric (Papineni et. Al 2002) and evaluated by human evaluators which were a monolingual non-expert translators have knowledge of only target language. Where BLEU score range between 0 > 1 and for manually checking we have set four parameters as shown below.

TABLE III. Manually Evaluation Scores

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

For BLEU score based evaluation, one target translation reference has been used to calculate a score which was prepared by same linguistic experts those who prepared training data. For incremental training, all training data was collected from BBC Urdu website. The system has been evaluated after every 100 training documents. BLEU scores for per domain shown in chart 1 to chart 5.

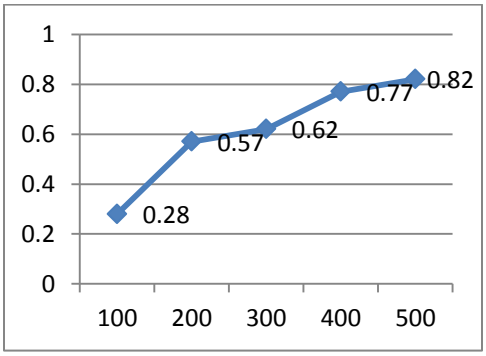

Chart 1: Political News Accuracy

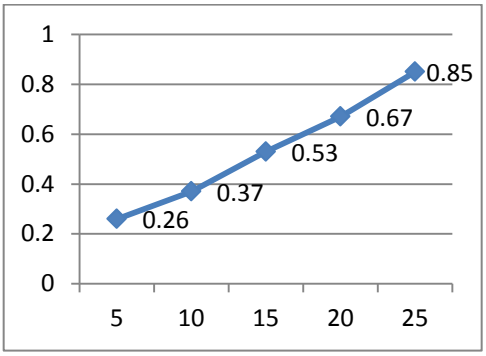

Chart 2: Tourism News Accuracy

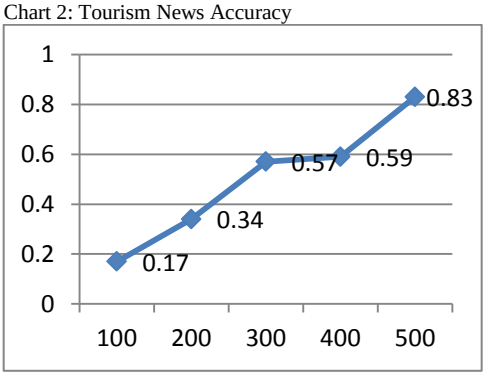

Chart 3: Entertainment News Accuracy

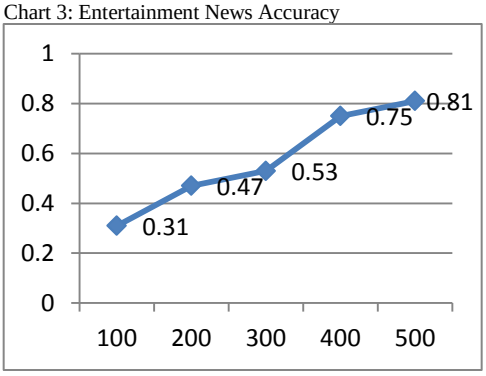

Chart 4: Sports News Accuracy

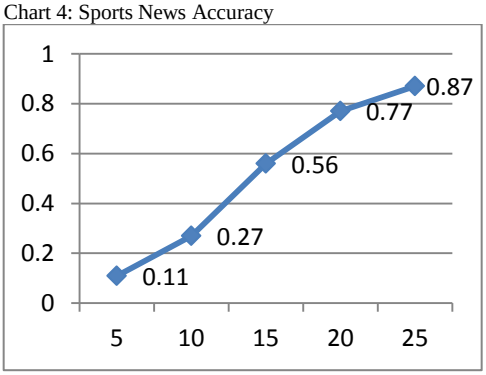

Chart 5: Health News Accuracy

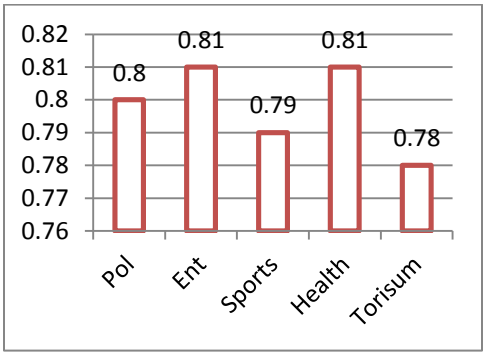

Chart 6: Overall Accuracy without Text Classifier

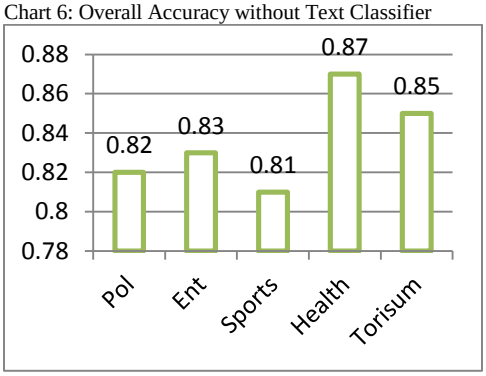

Chart 7: Overall Accuracy with Text Classifier

Manual testing was performed at the end of the training section. Test set contained 10 documents from each domain combined 1123 sentences. In manual testing 85% sentences got score 3 and 2 and 10% sentences got score 1 and remaining got score 0 which are new to the system and overall BLEU score was 0.86 for the same set of sentences. The text classifier before translation showed an increase in overall accuracy. The text classifier helped translation algorithm to pick correct translations phrases according to the domain of input text. The text classifier was evaluated using standard metrics as shown below.

Manual testing was performed at the end of the training section. Test set contained 10 documents from each domain combined 1123 sentences. In manual testing 85% sentences got score 3 and 2 and 10% sentences got score 1 and remaining got score 0 which are new to the system and overall BLEU score was 0.86 for the same set of sentences. The text classifier before translation showed an increase in overall accuracy. The text classifier helped translation algorithm to pick correct translations phrases according to the domain of input text. The text classifier was evaluated using standard metrics as shown below.

$$Recall = \frac{c_{ii}}{\sum_j c_{ij}} \quad (21)$$

$$Precision = \frac{c_{ii}}{\sum_j c_{ji}} \quad (22)$$

$$Accuracy = \frac{\sum_i c_{ii}}{\sum_j \sum_i c_{ij}} \quad (23)$$

TABLE IV. CONFUSION MATRIX OF TEXT CLASSIFIER

Documents	Assigned to Political	Assigned to Entertainment	Assigned to Sports	Assigned to Tourism	Assigned to Health
Political	471	8	3	0	0
Entertainment	13	482	7	0	0
Sports	14	6	487	0	0
Tourism	2	4	3	25	0
Health	0	0	0	0	25

TABLE V. PER CLASS RECALL AND PRECISION

	Recall	Precision
Political	0.977	0.942
Entertainment	0.960	0.964
Sports	0.960	0.974
Tourism	0.735	1
Health	1	1

The text classifier able to classify any given text document with overall accuracy 0.961. The text classifier was failed when document did not contain sufficient text to classify or text was very ambiguous for classifier like a political document which contains more sports related text than politics.

Our experiment shows that simple statistical model like HMM also yields good results for the closely related language pair. HMM based model quite popular in the field of part of speech (POS) tagging and Named Entity (NE) tagging and researcher showed really good results for sequence tagging NLP applications. Various researchers [Thorsten Brants, 200] had been shown that with a good amount of training tokens even simple statistical model also perform well compared to MaxEnt etc.

Appendix 3 shows that sample output and comparison of Google translator and our machine translation system. The proposed system generates nearly perfect or perfect translation of given text compared to Google translator which generates grammatical incorrect, meaningless and partial output in all cases. The system's output was compared with all five domains. Urdu inputs examples were quite simple without any ambiguous words.

The comparison is difficult between both systems because both systems used different training data sets, but we had checked the entire words list manually on Google translator and nearly all words were in its translation database, but decoder was not able to translate the input text by using its knowledge base. Google translator has very rich phrase translation database but the translation is still quite poor for Urdu-Punjabi language pair.

VI. CONCLUSION

The Paper has presented incremental learning based Urdu to Punjabi machine translation system. In place of parallel corpus, where system learns parameters from parallel sentences of source and target language. The proposed system used manually mapped parallel phrases training data and learned the parameters for translation model and language model rather than using parallel sentences corpus. In preprocessing phase, the system has used rules for segmentation, tokenization and text classification system to translate given text according to a preferred domain which also helped translation system to improve overall accuracy. The system has been trained and tested for Urdu Punjabi language pair which is closely related languages and share grammatical structure and vocabulary. Urdu and Punjabi languages are resources-poor languages and one should need a huge amount of parallel corpus to train statistical machine translation model to get decent accuracy. In our learning method, the system has able to achieve 0.86 BLEU score which is relatively good compared to other statistical translation systems. Like Urdu and Punjabi, many other Asian languages are resource poor languages and this approach can be applied straight away for other closely related language pairs.

ACKNOWLEDGEMENT

We are thankful to Technology Development for Indian Languages (TDIL) for supporting us and providing the Urdu Punjabi parallel corpus of Health and Tourism domain which is used for the evaluation and comparison.

REFERENCES

[1] Brants, Thorsten, TnT -- A Statistical Part-of-Speech Tagger, In proceeding of the 6th Applied NLP Conference, pages 224-231 2000

[2] Chen and Goodman 1998, "An Empirical Study of Smoothing Techniques for Language Modeling

[3] Durrani, Nadir et.al. "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

[4] Durrani, Nadir et.al "Hindi-to-Urdu Machine Tarnation through transliteration" Processing of the 48th Annual Meeting of the Association for Computational Linguistics, pages 465-474 2010

[5] Goyal, Vishal et.al, Evaluation of Hindi Punjabi Machine Translation System, IJCSI International Journal of Computer Science Issues, Vol. 4 No. 1, pages: 36-39 2009

[6] Garje G V, Survey of Machine Translation Systems in India, International Journal on Natural language Computing Vol 2, No4, pages: 47-67 Oct 2013

[7] Josan, Gurpreet Singh, A Punjabi To Hindi Machine Translation System, Companion volume – Posters and Demonstration, pages: 157-160 2008

[8] Kansal, Rohit et.al, "Rule Based Urdu Stemmer", processing of Colling 2012, Demonstration paper, pages 267-276 2012

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

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

[11] L. Rabiner, A Tutorial on Hidden Markov Model and Selected Application in Speech Recognition, in Proceeding on the IEEE, Vol. 77, Issue. 2, pages: 257-286 1989

[12] Papinni, Kishore, (2002), Bleu: a method for automatic evaluation of machine translation, ACL-2002: 40th Annual meeting of the Association for Computational Linguistics. pages 311–318

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

[14] Sani, Tajinder Singh (2011) "Word Disambiguation in Shahmukhi to Gurmukhi Transliteration", Processing of the 9th Wordshop on Asian Language Resources, Chiang Mai, Thailand, November 12 and 13, 2011 pages: 79-87

APPENDIX 1 (TRANSLATION MODEL TRAINING FILE)

Training File	
1	پاکستانی کرکٹ ٹیم|||[ਪਾਕਿਸਤਾਨੀ ਕ੍ਰਿਕੇਟ ਟੀਮ]
2	نیوزی لینڈ کے خلاف|||[ਨਿਊਜੀਲੈਂਡ ਦੇ ਖਿਲਾਫ]
3	پہلا|||[ਪਹਿਲਾ]
4	میچ|||[ਮੈਚ]
5	31|||[31]
6	جنوری کو|||[ਜਨਵਰੀ ਨੂੰ]
7	ویلنگٹن|||[ਵੈਲਿੰਗਟਨ]
8	میں|||[ਵਿੱਚ]
9	جبکہ|||[ਜਦੋਂ ਕਿ]
10	دوسرا|||[ਦੂਜਾ]
11	تین|||[ਤਿੰਨ]
12	فروری کو|||[ਫਰਵਰੀ ਨੂੰ]
13	نیپیئر|||[ਨੇਪੀਅਰ]
14	کھیلے میں|||[ਵਿੱਚ ਖੇਡੇ]
15	گی|||[ਗਈ]
16	-|||[।]
17	پہلا|||[ਪਹਿਲਾ]
18	میچ|||[ਮੈਚ]
19	نو|||[ਨੌਂ]
20	فروری کو|||[ਫਰਵਰੀ ਨੂੰ]
21	بنگلہ دیش کے|||[ਬਾਂਗਲਾਦੇਸ਼ ਦੀ]
22	خلاف|||[ਖਿਲਾਫ]
23	اور دوسرا|||[ਅਤੇ ਦੂਜਾ]
24	میچ|||[ਮੈਚ]
25	11|||[11]
26	فروری کو|||[ਫਰਵਰੀ ਨੂੰ]
2	انگلینڈ|||[ਇੰਗਲੈਂਡ]
28	کے خلاف|||[ਦੇ ਖਿਲਾਫ]
29	کھیلا|||[ਖੇਡਿਆ]
30	جائے گا|||[ਜਾਵੇਗਾ]
31	-|||[।]
32	دلچسپ بات یہ|||[ਦਿਲਚਸਪ ਗੱਲ ਇਹ]
33	ہے کہ کپتان|||[ਹੈ ਕਿ ਕਪਤਾਨ]
34	مصباح الحق آسٹریلیا|||[ਮਿਸਬਾਹ ਉਲ ਹੱਕ ਆਸਟਰੇਲਿਆ]
35	میں دو ٹیسٹ|||[ਵਿਚ ਦੋ ਟੇਸਟ]
36	میچ|||[ਮੈਚ]
37	کھیلے|||[ਖੇਡਣੇ]
38	ہیں|||[ਹਨ]

Training file with probability distribution		
1	پاکستانی کرکٹ ٹیم|||[ਪਾਕਿਸਤਾਨੀ ਕ੍ਰਿਕੇਟ ਟੀਮ]	1.0
2	نیوزی لینڈ کے خلاف|||[ਨਿਊਜੀਲੈਂਡ ਦੇ ਖਿਲਾਫ]	0.89
3	پہلا|||[ਪਹਿਲਾ]	0.90
4	میچ|||[ਮੈਚ]	1.0
5	31|||[31]	1.0
6	جنوری کو|||[ਜਨਵਰੀ ਨੂੰ]	1.0
7	ویلنگٹن|||[ਵੈਲਿੰਗਟਨ]	1.0
8	میں|||[ਵਿੱਚ]	0.76
9	جبکہ|||[ਜਦੋਂ ਕਿ]	0.98
10	دوسرا|||[ਦੂਜਾ]	0.64
11	تین|||[ਤਿੰਨ]	0.90
12	فروری کو|||[ਫਰਵਰੀ ਨੂੰ]	1.0
13	نیپیئر|||[ਨੇਪੀਅਰ]	1.0
14	کھیلے میں|||[ਵਿੱਚ ਖੇਡੇ]	1.0
15	گی|||[ਗਈ]	0.34
16	-|||[।]	1.0
17	پہلا|||[ਪਹਿਲਾ]	0.90
18	میچ|||[ਮੈਚ]	1.0
19	نو|||[ਨੌਂ]	0.98
20	فروری کو|||[ਫਰਵਰੀ ਨੂੰ]	1.0
21	بنگلہ دیش کے|||[ਬਾਂਗਲਾਦੇਸ਼ ਦੀ]	0.89
22	خلاف|||[ਖਿਲਾਫ]	0.76
23	اور دوسرا|||[ਅਤੇ ਦੂਜਾ]	0.89
24	میچ|||[ਮੈਚ]	1.0
25	11|||[11]	1.0
26	فروری کو|||[ਫਰਵਰੀ ਨੂੰ]	1.0
27	انگلینڈ|||[ਇੰਗਲੈਂਡ]	1.0
28	کے خلاف|||[ਦੇ ਖਿਲਾਫ]	0.68
9	کھیلا|||[ਖੇਡਿਆ]	0.87
30	جائے گا|||[ਜਾਵੇਗਾ]	0.89
31	-|||[।]	1.0
32	دلچسپ بات یہ|||[ਦਿਲਚਸਪ ਗੱਲ ਇਹ]	0.78
33	ہے کہ کپتان|||[ਹੈ ਕਿ ਕਪਤਾਨ]	1.0
34	مصباح الحق آسٹریلیا|||[ਮਿਸਬਾਹ ਉਲ ਹੱਕ ਆਸਟਰੇਲਿਆ]	1.0
35	میں دو ٹیسٹ|||[ਵਿਚ ਦੋ ਟੇਸਟ]	0.98
36	میچ|||[ਮੈਚ]	1.0
37	کھیلے|||[ਖੇਡਣੇ]	0.97

APPENDIX 2 (PHRASE SEPARATED FILE TO GENERATE LANGUAGE MODEL)

[ਪਾਕਿਸਤਾਨੀ ਕ੍ਰਿਕੇਟ ਟੀਮ] [ਆਪਣੇ] [ਸਭ ਤੋਂ ਵੱਡੇ] [ਇਮਤਿਹਾਨ] [ਦੇ ਲਈ] [ਮੰਗਲਵਾਰ] [ਦੀ ਰਾਤ] [ਕ੍ਰਿਕੇਟ ਦੇ ਵਿਸ਼ਵ ਕੱਪ] [ਦੀ ਯਾਤਰਾ ਤੇ] [ਰਵਾਨਾ ਹੋ] [ਰਿਹਾ ਹੈ] [।] [ਇਸ] [ਸਫਰ] [ਵਿੱਚ ਓਹ] [ਪਹਿਲਾ] [ਪੜਾਵ] [ਨਿਊਜੀਲੈਂਡ ਵਿੱਚ] [ਪਾਏ] [ਗਈ] [ਜਿੱਥੇ] [ਉਸਨੂੰ] [ਨਿਊਜੀਲੈਂਡ ਦੇ ਖਿਲਾਫ] [ਦੋ] [ਇੱਕ ਰੋਜਾ] [ਅੰਤਰਰਾਸ਼ਟਰੀ] [ਮੈਚ] [ਖੇਡਣੇ] [ਹਨ ਜਿਸਦੇ] [ਬਾਅਦ ਉਹ] [ਆਸਟਰੇਲਿਆ] [ਪਹੁੰਚ] [ਜਾਵੇਗੀ] [।]

[ਪਾਕਿਸਤਾਨੀ ਕ੍ਰਿਕੇਟ ਟੀਮ] [ਨਿਊਜੀਲੈਂਡ ਦੇ ਖਿਲਾਫ] [ਪਹਿਲਾ] [ਮੈਚ] [11] [ਜਨਵਰੀ ਨੂੰ] [ਵੈਲਿੰਗਟਨ] [ਵਿੱਚ] [ਜਦੋਂ ਕਿ] [ਦੂਜਾ] [ਤਿੰਨ] [ਫਰਵਰੀ ਨੂੰ] [ਨੇਪੀਅਰ] [ਵਿੱਚ ਖੇਡੇ] [ਗਈ] [।]

[ਪਾਕਿਸਤਾਨੀ ਟੀਮ] [ਵਿਸ਼ਵ ਕੱਪ ਤੋਂ] [ਪਹਿਲਾਂ] [ਸਿਡਨੀ] [ਵਿੱਚ ਦੋ] [ਵਾਰਮ ਅੱਪ] [ਮੈਚ] [ਵੀ] [ਖੇਡੇ] [ਗਈ] [।]

Bigram Phrase based Language Model

Bigram Phrases: $count(phrase_{i-1}, phrase_i)$	Count
ਪਾਕਿਸਤਾਨੀ ਕ੍ਰਿਕੇਟ ਟੀਮ, ਆਪਣੇ	10
ਆਪਣੇ, ਸਭ ਤੋਂ ਵੱਡੇ	8
ਸਭ ਤੋਂ ਵੱਡੇ, ਇਮਤਿਹਾਨ	1
ਇਮਤਿਹਾਨ, ਦੇ ਲਈ	1
ਦੇ ਲਈ, ਮੰਗਲਵਾਰ	2
ਮੰਗਲਵਾਰ, ਦੀ ਰਾਤ	2
ਦੀ ਰਾਤ, ਕ੍ਰਿਕੇਟ ਦੇ ਵਿਸ਼ਵ ਕੱਪ	3

Bigram Word based Language Model

Bigram Words: $count(w_{i-1}, w_i)$	Count
ਪਾਕਿਸਤਾਨੀ, ਕ੍ਰਿਕੇਟ	123
ਕ੍ਰਿਕੇਟ ਟੀਮ	421
ਟੀਮ, ਆਪਣੇ	321
ਆਪਣੇ, ਸਭ	131
ਸਭ, ਤੋਂ	634
ਤੋਂ, ਵੱਡੇ	102
ਵੱਡੇ, ਇਮਤਿਹਾਨ	1

APPENDIX 3

Output comparison of all domain. Mistakes are underlined.

Political Input Text	پاکستان کی بّری فوج کے سربراہ جنرل راحیل شریف نے کہا ہے کہ سکیورٹی فورسز چین پاکستان اقتصادی راہداری منصوبے کے خلاف چلائی جانے والی تمام مہمات سے آگاہ ہیں اور اس شاندار منصوبے کو حقیقت کا رنگ دینے کے لیے کسی قربانی سے بھی دریغ نہیں کریں گی۔
Our Translator output	ਪਾਕਿਸਤਾਨ ਦੀ ਫੋਜ ਦੇ ਪ੍ਰਮੁੱਖ ਜਨਰਲ ਰਅਹੀਲ ਸ਼ੀਫ ਨੇ ਕਿਹਾ ਹੈ ਕਿ ਸੁਰੱਖਿਆ ਬਲਾਂ ਚੀਨ ਪਾਕਿਸਤਾਨ ਆਰਥਕ ਰਾਹਦਾਰੀ ਯੋਜਨਾ ਦੇ ਖਿਲਾਫ ਲੀਕੇ ਜਾਣ ਵਾਲੀਆਂ ਸਭ ਮੁਹਿੰਮਾ ਦੀ ਜਾਣੂ ਹਨ ਅਤੇ ਇਸ ਸ਼ਾਨਦਾਰ ਯੋਜਨਾ ਨੂੰ ਸਚਾਈ ਦਾ ਰੰਗ ਦੇਣ ਲਈ ਕਿਸੇ ਕੁਰਬਾਨੀ ਤੋਂ ਵੀ ਗੁਰੇਜ ਨਹੀਂ ਕਰਨ ਗਏ ।
Google Translator output	ਫੌਜ ਸਟਾਫ ਜਨਰਲ ਰਾਹੀਲ ਸ਼ਰੀਫ ਦੇ ਪਾਕਿਸਤਾਨ ਦੇ ਪ੍ਰਧਾਨ ਨੇ ਕਿਹਾ ਹੈ ਕਿ ਪਾਕਿਸਤਾਨ ਦੇ ਸੁਰੱਖਿਆ ਬਲ, ਚੀਨ ਲਈ ਸਭ ਮੁਹਿੰਮ ਆਰਥਿਕ ਕੋਰੀਡੋਰ ਪ੍ਰਾਜੈਕਟ ਨੂੰ ਦੇ ਖਿਲਾਫ ਸ਼ੁਰੂ ਕੀਤੀ ਹੈ ਅਤੇ ਇਸ ਤੱਥ ਦੇ ਜਾਣੂ ਹਨ ਬਲੀ ਨਾ ਕਰੇਗਾ, ਜੋ ਕਿ ਇਸ ਸ਼ਾਨਦਾਰ ਪ੍ਰਾਜੈਕਟ ਦੀ ਰੰਗ ਹੈ.
Health Input Text	منہ کے اندرونی حصے کی مختلف بیماریوں میں دانتوں کا کم زور ہو جانا، کیڑا لگنا، دانتوں کا ٹیڑھا ہو جانا اور مسوڑھوں کا انفیکشن عام ہے۔ دانتوں اور مسوڑھوں کی زیادہ تر بیماریوں کی علامات فوراً ظاہر نہیں ہوتیں، اور شدید تکلیف کا باعث بنتی ہیں۔
Our Translator output	ਮੁੰਹ ਦੇ ਅੰਦਰੂਨੀ ਹਿੱਸੇ ਦੀ ਵਿਭਿੰਨ ਬਿਮਾਰੀਆਂ ਵਿੱਚ ਦੰਦਾਂ ਦਾ ਕੰਮਜੋਰ ਹੋ ਜਾਣਾ , ਕੀੜਾ ਲੱਗਣਾ , ਦੰਦਾਂ ਦਾ ਟੇਢਾ ਹੋ ਜਾਣਾ ਅਤੇ ਮਸੂੜਿਆਂ ਦੀ ਇੰਫੈਕਸ਼ਨ ਆਮ ਹੈ । ਦੰਦਾਂ ਅਤੇ ਮਸੂੜਿਆਂ ਦੀਆਂ ਜ਼ਿਆਦਾਤਰ ਬਿਮਾਰੀਆਂ ਦੇ ਲਕਸ਼ਨ ਤੁਰੰਤ ਪ੍ਰਗਟ ਨਹੀਂ ਹੁੰਦੇ , ਅਤੇ ਗੰਭੀਰ ਤਕਲੀਫ ਦਾ ਕਾਰਨ ਬਣਦੀਆਂ ਹਨ ।
Google Translator output	ਮੂੰਹ ਤਣਾਅ ਨੂੰ ਘੱਟ ਕਰਨ ਲਈ, ਕੀੜਾ ਲੱਗਦਾ ਹੈ, ਟੇਢੇ ਦੰਦ ਅਤੇ ਗਮ ਲਾਗ ਜਾਣ ਦੇ ਅੰਦਰ ਦੰਦ ਦੇ ਵੱਖ-ਵੱਖ ਰੋਗ ਵਿੱਚ ਆਮ ਹਨ. ਜ਼ਿਆਦਾਤਰ ਦੰਦ ਅਤੇ ਗੰਮ ਦੀ ਬਿਮਾਰੀ ਦੇ ਲੱਛਣ ਤੁਰੰਤ ਵਿਖਾਈ ਨਾ ਕਰੇ, ਅਤੇ ਗੰਭੀਰ ਦਰਦ ਦਾ ਕਾਰਨ ਬਣ.
Entertainment Input Text	کبھی جڑواں بھائیوں میں فرق ظاہر کرنے کے لیے ایک مونچھ یا تل سے فرق دکھانے والے میک اپ آرٹسٹ اب پورے چہرے کا میک اپ ہی الگ طریقے سے ڈیزائن کرتے ہیں۔

Our Translator output	ਕਦੇ ਜੁੜਵਾਂ ਭਰਾਵਾਂ ਵਿੱਚ ਫਰਕ ਦਿਖਾਉਣ ਲਈ ਇੱਕ ਮੁੱਛਾ ਜਾਂ ਤਿਲ ਨਾਲ ਫਰਕ ਵਿਖਾਉਣ ਵਾਲੇ ਮੇਕਅੱਪ ਆਰਟਿਸਟ ਹੁਣ ਪੂਰੇ ਚਿਹਰੇ ਦਾ ਮੇਕਅੱਪ ਹੀ ਅਲੱਗ ਤਰਾਂ ਨਾਲ ਡੀਜਾਈਨ ਕਰਦੇ ਹਾਂ ।
Google Translator output	Twin ਬਣਤਰ ਕਲਾਕਾਰ, ਜੋ ਕਿ ਇੱਕ ਹੋ◌ੇਏ ਜ ਤਿਲ ਨੂੰ ਦਿਖਾਉਣ ਲਈ ਭਰਾ ਸਾਰੀ ਚਿਹਰੇ ਨੂੰ ਵੱਖ ਵੱਖ ਢੰਗ ਨੂੰ ਤਿਆਰ ਕਰ ਰਹੇ ਹਨ ਫ਼ਰਕ ਕਦੇ ਹੋਵੇਗਾ.
Tourism Input Text	لاطینی امریکہ کے ملک ارجنٹینا میں ساحل سمندر پر جانے والے ان افراد پر سخت نکتہ چینی کی جاری ہے جنھوں نے ناپید ہونے والی نسل کی ایک ڈولفن کے ساتھ سیلفی لینے کے لیے اسے سمندر سے باہر نکال لیا.
Our Translator output	ਲੈਟੀਨ ਅਮਰੀਕਾ ਦੇ ਦੇਸ਼ ਅਰਜਨਟੀਨਾ ਵਿੱਚ ਸਮੁੰਦਰੀ ਤੱਟ ਉੱਤੇ ਜਾਣ ਵਾਲੇ ਉਨ੍ਹਾਂ ਲੋਕਾਂ ਦਾ ਕਠੋਰ ਵਿਰੋਧ ਜਾਰੀ ਹੈ ਜਿਨ੍ਹਾਂ ਨੇ ਲੁਪਤ ਹੋਣ ਵਾਲੀ ਨਸਲ ਦੀ ਇੱਕ ਡਾੱਲਫਿਨ ਦੇ ਨਾਲ ਸੈਲਫ਼ੀ ਲੈਣ ਲਈ ਉਸਨੂੰ ਸਮੁੰਦਰ ਤੋਂ ਬਾਹਰ ਕੱਢ ਲਿਆ ।
Google Translator output	ਨਸਲ, ਜੋ ਜਿਹੜੇ ਬੀਚ 'ਤੇ ਜਾਣ ਦੀ ਆਲੋਚਨਾ ਕੀਤੀ ਹੈ ਦੇ ਸਵੈ-ਤਬਾਹ ਲੈ ਲਈ ਇੱਕ ਡਾਲਫਿਨ ਨਾਲ ਲਾਤੀਨੀ ਅਮਰੀਕੀ ਦੇਸ਼ ਵਿਚ ਅਰਜਨਟੀਨਾ ਸਮੁੰਦਰ ਨੂੰ ਉਸ ਨੂੰ ਬਾਹਰ ਲੈ ਗਿਆ.
Sports Input Text	بھارتی کرکٹ بورڈ نے کہا ہے کہ قومی کرکٹ ٹیم کے کپتان مہندر دھونی پیر کو معمول کی تربیت کے دوران کمر کے درد میں مبتلا ہونے کے بعد ایشیا کپ میں ٹیم کا حصہ نہیں ہوں گے۔
Our Translator output	ਭਾਰਤੀ ਕ੍ਰਿਕੇਟ ਬੋਰਡ ਨੇ ਕਿਹਾ ਹੈ ਕਿ ਰਾਸ਼ਟਰੀ ਕ੍ਰਿਕੇਟ ਟੀਮ ਦੇ ਕਪਤਾਨ ਮਹਿੰਦਰ ਧੋਨੀ ਸੋਮਵਾਰ ਨੂੰ ਰੁਟੀਨ ਸਿਖਲਾਈ ਦੇ ਦੌਰਾਨ ਕਮਰ ਦੇ ਦਰਦ ਤੋਂ ਪੀੜਿਤ ਹੋਣ ਦੇ ਬਾਅਦ ਏਸ਼ਿਆ ਕੱਪ ਵਿੱਚ ਟੀਮ ਦਾ ਹਿੱਸਾ ਨਹੀਂ ਹੋਣਗੇ ।
Google Translator output	ਭਾਰਤੀ ਕ੍ਰਿਕਟ ਟੀਮ ਦੇ ਕਪਤਾਨ ਮਹਿੰਦਰ ਸਿੰਘ ਧੋਨੀ ਨੇ ਕਿਹਾ ਹੈ ਕਿ ਸੋਮਵਾਰ ਨੂੰ ਰੁਟੀਨ ਦੀ ਸਿਖਲਾਈ ਦੌਰਾਨ ਪਿੱਠ ਦੇ ਦਰਦ ਨਾਲ ਪੀੜਤ ਦੇ ਬਾਅਦ, ਏਸ਼ੀਆਈ ਕੱਪ 'ਚ ਟੀਮ ਦਾ ਹਿੱਸਾ ਨਾ ਹੋਵੇਗਾ.