A Model for English to Urdu and Hindi Machine Translation System using Translation Rules and Artificial Neural Network

Shahnawaz Khan[1] and Imran Usman[2]
[1]Department of Information Technology, University College of Bahrain, Bahrain
[2]College of Computing and Informatics, Saudi Electronic University, Saudi Arabia

Abstract: *This paper illustrates the architecture and working of a proposed multilingual machine translation system which is able to translate from English to Urdu and Hindi. The system applies translation rules based approach with artificial neural network.The efficient pattern matching and the ability of learning by examples makes neural networks suitable for implementation of a translation rule based machine translation system.This paper also describes the importance of machine translation systems and status of the languages in a multilingual country like India.Machine translation evaluation score for translation output obtained from the system has been calculated using various methods such as n-gram bleu score, F-measure, Meteor and precision, recall. The evaluation scores achieved by the system for around 500 Hinditest sentences are as: n-gram bleu score 0.5903; Metric for Evaluation of Translation with Explicit ORdering (METEOR) score achieved is 0.7956 and F-score of 0.7916 and for Urdu n-gram bleu score achieved by thesystem is 0.6054; METEOR score achieved is 0.8083 and F-score of 0.8250.*

Keywords: *Machine translation, artificial neural network, english, hindi, urdu.*

Received September 19, 2015; accepted June 8, 2016

1. Introduction

According to Nirenbung [11], machine translation is the process by which a computer must be able to produce the equivalent natural language text (such as Hindi or Urdu)as output from a given source language text (such as English) using computer software in such a way so that the meaning of the target language text is same as that of the source language text. Machine Translation is defined as translation of text from one natural language to another using computer [4]. Machine Translation (MT) is in great demand now-a-days due to globalization of information.Information needs to be accessed from different parts of the world. Most of this information is available in English only. There is a great number of people around the world who do not understand English. Therefore, these people are not able to grasp all the information available. The aim of building a machine translation system is to overcome language barriers to some extent.

Machine Translation has been the area of interest since 1950s with the Georgetown University and International Business Machines (IBM) experiment of automatic translation of over 60 Russian sentences in organic chemistry domain. In this experiment, system contained only six grammar rules and around 250 items vocabulary. The successful demonstration of the experiment gained worldwide attention. The earliest installations of machine translation systems were in military translation services and governmental [13] primarily because of the cost of the required computer hardware. A large community of researchers and organizations are working in the area of machine translation and natural language processing these days. According to [4], translation output produced by MT systems and translation tools can be divided into four basic types: translation of publishable quality, translation to get the essential contents of the text being translated, translation for one to one communication and translation for information extraction, information retrieval and database access etc. within the multilingual systems. A high quality fully automatic machine translation appears to require an artificial intelligence equivalence to human intelligence. In this paper, we are not apprehensive about the high quality fully automatic machine translation of unrestricted text, but rather building an MT system that can overcome linguistic barriers in one way or another.

The MT system demonstrated in this paper has been implemented using artificial neural network and translation rules. Neural networks are very efficient in pattern matching and have the ability of learning by examples. Artificial neural network and rule based technique have been used for development of the MT systems such as in Parallel Runtime Scheduling and Execution Controller (PARSEC)[5], JANUS [23], English to Arabic [1] and English to Urdu MT System [15], English to Sanskrit MT system[10]. Rule based

MT approach belongs to the classical approaches of machine translation. Rule based MT approach has been implemented by some of the most popular MT systems such as Systran [18] and Eurotra [6].

This paper has been divided into five sections. Next section represents the status of languages (English, Hindi and Urdu) and grammatical similarity between Hindi and Urdu. Section three describe the architecture of our system and discusses translation rules and encoding-decoding process. Section four discusses the results obtained from the system output. Last section concludes this paper with our ongoing work and future work plans.

2. The status of Languages: Urdu, Hindi and English

Ethnologue [7] catalogs around 6900 known living languages spoken around the world and according to Ethnologue research, it came out that around 6% (i.e., 389) languages are spoken by 94% population of the world. Globalization, international businesses and World Wide Web has brought the world together. English is the most commonly used language for websites contents and other communications. English is used by 55.4% of all the websites as their content language. Hindi and Urdu are merely used by less than 0.1% of the total websites[22]. All the people cannot access this information due to language barrier. Hindi is the official languages in India and in Fiji. Urdu is the official language in Pakistan and India (Jammu and Kashmir). Hindi is spoken by around 853 million speakers and Urdu is spoken by around 164 million speakers as their first and second languages in the world [22]. Hindi, as first language only, is spoken by 260 million speakers and 64 million speakers use Urdu as their first language [7]. English, in India, is used for government communication and notification. English is the topmost language for Internet and a huge amount of information is available in English. Average literacy level in India is 65.4%. In, India, there are less than 5 % people who can either write or read English. Over 95% of the population in India does not get benefited from English based information technology [21].

Hindi and Urdu are very close languages at phonological level and at grammatical level also [9, 14]. Both the languages follow similar sentence structure, verb morphology and complex verb predicates and same post-positions [16]. Urdu is written in a script which is a derivation of Persio-Arabic script and Hindi is written in Devanagiri script [8]. Urdu language's vocabulary has been borrowed from Persian and Arabic and Hindi language's vocabulary is based on Sanskrit [14]. Hindi and Urdu are Subject-Object-Verb (SOV) languages with respect to word order. In terms of branching, Hindi/Urdu is neither purely right-branching nor left-branching; phenomena of both forms can be found. Constituents order in the sentences as a whole lack of the "hard and fast" governing rules. Frequent deviations from the normative word position can be found, describable in terms of small number of rules, accounting for the facts beyond the pale of the label of "Subject-Object-Verb". The MT System demonstrated in this paper considers the SOV word order of both languages.

3. System Architecture and Implementation

The architecture of the proposed MT system model is shown in Figure 1 below. The model is based on neural network and translation rules approach. The Artificial Neural Network (ANN) model in Figure 1 has been trained on two typesof data: translation rules and bilingual dictionaries. Translation rules have been created for transferring the grammatical structure of the source language sentences into the target language sentences. These rules are encoded for neural network training. A neural network object is created after training which is accessed by the system on runtime for retrieving the suitable translation rule of the sentence being translated. ANN model has been trained on bilingual dictionaries for English-Hindi and English-Urdu language pairs. Tokens in bilingual dictionaries do not only contain meanings of the source language words but also have been attached with semantic information associated with the words to build the knowledgeable dictionaries.

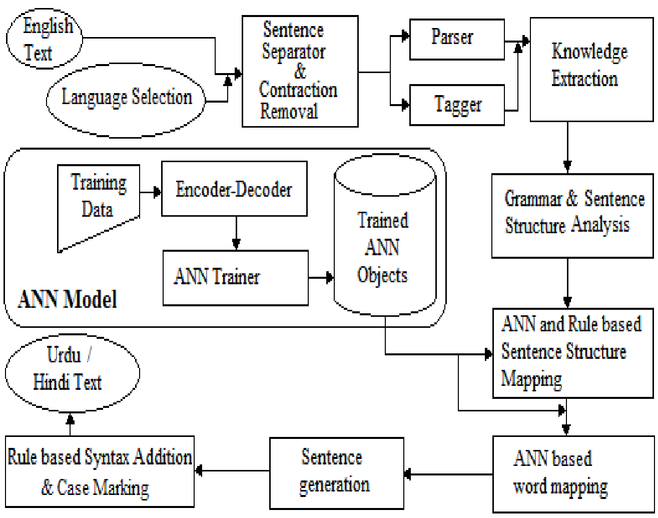

Figure 1. System architecture.

When the text being translated is given as input to the system, it is processed for contractions removal after which the text is split into sentences and these sentences are then parsed and tagged with Stanford typed dependency parser [3] and Stanford maximum entropy tagger [19]. Parsed and tagged sentences are processed for semantic information extraction. The sentence is parted into constituents (such as subject, object, verb etc.,) and a grammar structure of the sentences is generated. These structures are encoded to form the input query for ANN trained objects of translation rules in ANN model which returns target language grammatical structure for the sentences. The return rules are decoded to form the target language's sentence structure. All the constituents of the source language sentence are transformed in same fashion from the ANN model and are decoded. These constituents are translated with the help of ANN models of bilingual dictionaries and Encoder and Decoder.

3.1. Artificial Neural Network Model

The ANN based training module of our proposed technique comprises of back propagation neural network with Levenberg-Marquardt (LM) algorithm [17]. Initially, the training set constitutes of tokens of a language to be recognized by their target mapping-language outputs. The objective of the central ANN based classifier is to map the right set of translated words using an LM based back propagation neural network. The training data is first transformed into a set of data which is quantifiable so that it can be passed on to the neural network. For this, we introduce ANN encoder/Decoder structures, as can be seen in Figure 1, which translates a given letter into its corresponding position value in the language alphabet set. As for an example, A=1, B=2, and so on. In order to place a bound on the limit and make it simple, we normalize these values between [0-1]. Once the input data is generated, we next repeat the same procedure on the target data and present them to the LM based ANN classifier.

LM provides fast and stable convergence and can be used in small and medium sized optimization problems. It blends steepest decent algorithm and Gaussian-Newton method by inheriting the stability of steepest decent method and speed superiority of Gaussian-Newton method. In the proposed technique, let us define the performance measure $F(w)$ to be the sum of squared errors between the networks output and the target output. Our goal is to minimize this error.

$$F(w) = \sum e \tag{1}$$

Where, e is the error vector and $w=[w_1,w_2,w_3,...w_n]$ are the weights. The increment in weights Δw can be obtained as:

$$\Delta \mathrm{w} = \left[\mathrm{J}^{\mathrm{T}}\mathrm{J} + \mu \mathrm{I}\right]^{-1} \mathrm{J}^{\mathrm{T}}\mathrm{e} \tag{2}$$

Where, μ is the learning rate momentum and J is the Jacobian matrix. We use a decay rate $0<\delta<1$ to control the learning rate such that it can avoid being trapped into the local minima. In order to do so, whenever $F(w)$ decreases, we multiply δ and μ. On the other hand, if $F(w)$ increases, μ is divided by δ.

For the sake of generality, and for the sake of understanding, the standard LM training algorithm can be depicted in the following pseudo code.

- *Step 1*: Initialize all the weights and the parameter μ value.
- *Step 2*: Compute the sum of errors using Equation (1).
- *Step 3*: Find the change in weights using Equation (2).
- *Step 4*: Re-compute $F(w)$ using $w+\Delta w$

keeping the following condition this time:

IF $F(w) < F(w)$ in step 2,
THEN $\mu = \mu.\delta$, goto step 2
ELSE $\mu = \mu / \delta$, goto step 4
ENDIF

3.2. Encoder Decoder

A datasets of translation rules and bilingual dictionaries for English-Hindi and English-Urdu language pairs has been created. English letters have been used to represent Hindi, Urdu and English text. Each English alphabet is represented (a =1, b =2 …) by five bits (as there are 26 alphabets (2^4 =16 and 2^5 =32 so needs 5 bits). Value of each alphabet is converted to decimal by dividing 26 (a=1/26, b= 2/26…) to train the neural network. Words/tokens and translation rules are changed to a sequence of numbers to create dataset to train neural network. Encoder converts the grammar rules and token/words into numeric encoded form, a form which is suitable for input for ANN models and Decoder converts the numeric coded grammar rules and token/words back to human readable form. To automate the process a java class was created for encoding training data in numeric form. Encoder java class converts training data into numeric form from a text file where data is present in human readable form. Numeric form is difficult to read by a human but easy for a program.

The system has been implemented using Java and Matlab. ANN models have been trained and created in Matlab. Encoder-Decoder module for creating datasets for training neural network is implemented in Java. Stanford Parser and Stanford Tagger are available in Java library form. System processes the output of tagger and parser in Java and implementation of all the modules except ANN models is Java based.

The input layer of grammatical structure network contains 42 nodes, hidden layer contains 100 nodes and output layer contains 30 nodes. Training error goal for mean squared error was set to 10^{-8} which was achieved after 29 epochs. Neural network has been trained for translation rules with a data set of around 465 input-output pair of grammar rules for each language pair. The neural network for knowledgeable bilingual dictionary has been trained with a data set of around 9000 input-output pair of each English-Urdu and English-Hindi words with associated semantic information. The input layer of bilingual dictionary network contains 10 nodes, hidden layer contains 100 nodes and output layer contains 32 nodes (for meaning and semantic information). Mean squared error goal was set to training error of 10^{-8} which was achieved after 333 epochs.

A java class encodes the tokens and linguistic rulesand sends the output to ANN model which queries the neural networks for mapping them to their equivalent target language tokens and linguistic rules. Neural network then maps these numeric values and produces equivalent results in numeric form which are then again passed to the java class which decodes numeric output retrieved from neural network back to human readable form with the help of decoder. Thesemantic informationattached with the word tokens is further processed and target language meaning and attached information is extracted. Suffix in the verb and marker with the subject are attached on the basis of semantic information obtained from the neural network and information obtained in the Grammar Analysis and Sentence Structure Recognition module. These parts are then arranged according to the grammar structure obtained from grammatical structure network and the output is presented in Romanized form.

3.3. Translation Rules

System uses translation rules created for various classes of the sentences. The system at the current stage is able to handle all forms (affirmative, negative and interrogative) of simple English languages sentences. The verbs and nouns in the output are inflected based up on the grammaticalinformation like tense, gender, number person etc. extracted in the knowledge extraction module. Translation rules for the following structures of the sentences have been written:

SV, SVSc, SVO, SVG, SVGO, SVIoO, SVIn, SVInIn, SVInO, SVpPO, SVpPOpPO, SVpPOpPOpPO, SVOpPO, SVOpPOpPO, SVOpPOpPOpPO; Where S=Subject, V=Verb, Sc=Subject Compliment, Io=Indirect Object, In= Infinitive, G=Gerund, p=preposition and PO =Prepositional Object.

Consider the following rule example for the following English sentence: “I lent my book to a friend.”Following translation rule will be used for the Urdu translation:

IF (Sentence structure is SVOpPO and tense is Past-Indefinite and sentence is affirmative in active voice)
THEN (Urdu grammar=subject (S)+object (O)+ prepositional object (PO) +preposition (P)+verb (V)).

Syntax addition: As direct object is present in the sentence so case marker ‘ne’ has to be added and marker ‘ā’ to verb will also be added in Urdu translation. This is decided on the basis of tense, sentence structure and coupled information (number, person, gender) with the Urdu meaning of the word.We have written translation rules for each tense considering all cases of person, number, gender, and person and sentence structure.The general structure for the grammar translation rule for training neural network as follows:

Input=gclass_tense_type_category_voice
Output=urdu/hindi grammar;
For example
Input=svoppo_pastInd_s_aff_act;
Output=s_o_po_p_v.

Where gclass is the grammar class of sentence like svo, tense is like Past Indefinite, type of the sentence is simple, complex, imperative etc., category is affirmative, interrogative etc. and voice is active or passive.Some examples of translation rules are as follows; we have chosen Hindi as the target language in the following examples:

English Sentence (E.S.): Dr.I.Usman is a researcher When system scans this type of sentences following rule fulfill the conditions.

Rule: If (sentence structure is SVSc and tense is present and affirmative sentence in active voice)
Then (Hindi grammar = S + Sc + V)

E.S.: Has the bell rung?
Rule:If (sentence structure is SV and tense is present perfect and verb interrogative sentence in active voice)
Then (Hindi grammar = kya + S + V)
E.S.: The boy hadn’t lost his pen.
Rule: If (sentence structure is SVO and tense is past perfect and negative sentence in active voice)
Then (Hindi grammar = S + O + negative word + V)
E.S.: Why does he not want to go to watch the movie?
Rule: If (sentence structure is SVInInO and tense is present Indefinite and interrogative-negative sentence in active voice)
Then (Hindi grammar = S+O+In^2 +question word + negation word+In^1+V).
E.S.: I lent my pen to my friend.
Rule:If (sentence structure is SVOpPO and tense is past Indefinite and interrogative-negative sentence in active voice)
Then (Hindi grammar = S+O+ PO +p +V).

4. Results and Discussion

Various methods have been employed for evaluating the quality of machine translation output. N-gram MT-evaluation score of the system output has been calculated using BiLingual Evaluation Understudy (BLEU) [12]. BLEU is an IBM-developed metric and uses modified n-gram precision to compare the candidate translation against reference translations. It takes the geometric mean of modified precision scores of the test corpus and then multiplies the result by exponential brevity penalty factor to generate the BLEU score. The *Bleu* score is calculated as:

$$\mathrm{BLEU} = \mathrm{BP}.\exp(\ w_n \log p_n) \tag{3}$$

Where *BP* is brevity panelty, p_n is the modified precision score. We use N=4 in our baseline and uniform weights
w_n = 1/N.

To find out the BLEU score, modified precision (p_n) score is calculated as follows Equation (4) below:

$$p_n = \frac{\Sigma_{C\in\{Candidates\}}\Sigma_{n-gram\in C}\, Count_{clip}\,(n-gram)}{\Sigma_{C'\in\{Candidates\}}\Sigma_{n-gram'\in C'}\, Count_{clip}\left(n-gram'\right)} \tag{4}$$

In the above Equation (4), C is the set of candidate translation sentences and C' is the set of reference sentences. $Count_{clip}$ is calculated as:

$$Count_{clip} = \min(Count, Max\_ref\_Count) \tag{5}$$

Brevity penalty is calculated using the following equation

$$BP = \begin{cases} 1 & if\ c > r \\ e^{\left(1-\frac{r}{c}\right)} & if\ c \le r \end{cases} \tag{6}$$

Where r is the length of reference and *c* is the length of candidate. The n-gram BLEU score obtained by the system was 0.5903 for English to Hindi and 0.6054 for English to Urdu.

Another method which has used for calculating MT-evaluation score is *F-Measure*. It is an MT evaluation metric developed at the New York University. The *F-measure* is defined as the harmonic mean of precision and the recall as:

$$F-measure = \frac{2 * precision * recall}{precision + recall} \tag{7}$$

Precision is, fraction of correct instances among those that algorithm believes to belong to relevant subset [20], calculated as:

$$P = {|X \cap Y|}\big/{|Y|} \tag{8}$$

Where *X* in the above equations is the set of reference items and *Y* is the set of candidate items.

Recall is fraction of correct instances among all instances that actually belong to relevant subset [20] and can be calculated as:

$$R = {|X \cap Y|}\big/{|X|} \tag{9}$$

Where *X* in the above equations is the set of reference items and *Y* is the set of candidate items.

Metric for Evaluation of Translation with Explicit Ordering (or METEOR) [2] is an MT evaluation metric which is developed at Carnegie Mellon University. The Meteor metric is based on the weighted harmonic mean of unigram precision $P = m/w_t$ and unigram recall $R = m/w_r$. Where m is number of unigram matches, w_t is the number of unigrams in candidate translation and w_r is the reference translation. F_{mean} is calculated by combining the recall and precision via a harmonic-meanthat places equal weight on precision and recall as follows:

$$F_{mean} = 2PR/(P+R) \tag{10}$$

This measure is for congruity with respect to single words but for considering longer n-gram matches then a penalty *p*, according to the following algorithm, is calculated for the alignment as:

$$p = 0.5\left(\frac{c}{u_m}\right)^3 \tag{11}$$

Where *c* is the number of chunks, and u_m is the number of unigrams that have been mapped. The more mappings there are, that are not adjacent in the references and the candidate sentences, the higher the penalty will be. Final Meteor-score (M-score) can be calculated as:

$$M = F_{mean}(1-p) \tag{12}$$

Some features of the MT systems' output can be evaluated automatically for example fluency. Fluency can be checked by n-gram analysis of available reference translations. Some features are not easy to evaluatesuch as meaning or sense of translation. It is hard to compare between two different Machine Translation algorithms objectively.Following is the sampleHindi output of the MT system for the sample of English text given as input.

- Sample English Text: Shyam Kumar Singh is a student. He lives in Nainital. Nainital offers you refreshing environment. He enjoys playing hockey. He likes singing. He went to the fare with his mother. He saw an old man at the shop. The old man was buying a ring for his wife from the shop. He bought a book for his sister. He met his uncle. He wanted to go to watch the magician show. They decided to watch the show.
- Output Hindi Translation: SHYĀM KUMAR SINGH ekchhātrahai | wah NAINITAL me rahtāhai | NAINITAL tumkotāzāvātāvarandetāhai | wahhaukikhelnapasandkaratāhai | wahgānāpasandkartāhai | wahapnemātākesāthmelāgayāthā | wahdukānparekboodhāādamīdekhā thā | boodhāādamīmelā se apnīpatnīkeliyeekańgūthīkharīdrahāthā | wahapnībahankeliyeekpustakkharīdāthā | wahapnechāchāse milāthā | vahjādūgarkātamāshādekhnejānāchāhate the | vetamāshādekhnāfaislākiyāthā |

The words that are not present in the dictionaries are printed as it is in the translation in capitals.

The comparative scores of different Machine Translation evaluation methods such as BLEU

(BiLingual Evaluation Understudy), METEOR (M), F-measure (F) scores, unigram Precision (P), unigram Recall (R) for fifteen randomly selected sentences of various classes are shown in Figure 2.

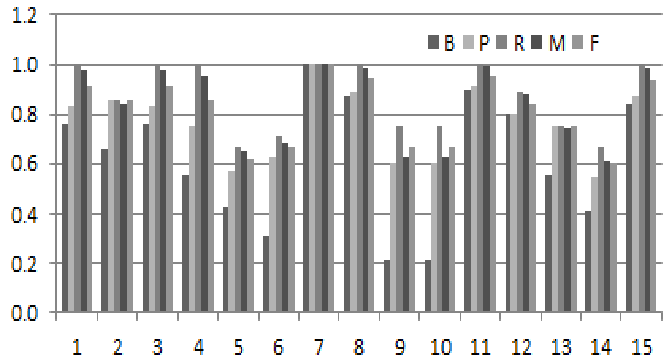

Figure 2. Comparative MT score for some sentences.

It has been seen from the results that system performs efficiently on those classes of sentences whose grammar rules are trained in the neural network. System depends on Stanford parser for typed dependency and Stanford tagger for part of speech tagging; if the parser or tagger makes an error for any sentence then same error will be propagated throughout the translation and will result in the wrong translation. We obtained an average n-gram bleu score 0.5903; METEOR score achieved is 0.7956 and F-score of 0.7916 and for Urdu n-gram bleu score achieved by the system is 0.6054; METEOR score achieved is 0.8083 and F-score of 0.8250.

5. Conclusions and Future Work

The translation results obtained from the system evaluated using machine translation evaluation methods and manually and it has been seen that the system works efficiently on the trained linguistic (translation) rules and bilingual dictionaries. Therefore, an enhancement to the grammar rules and size of bilingual dictionary will lead to the efficient and accurate machine translation system. Case marking is one of the important factors for the semantic accuracy of the translated text. In Hindi and Urdu, sentence meaning can change because of case markers only. We have also observed from the results of the system that if case marking is improved in the system, system will be able to produce more efficient results. The MT-evaluation scores, achieved by the system for around 500 test English sentences translated into Hindi and Urdu, are: n-gram bleu score 0.5903; METEOR score achieved is 0.7956 and F-score of 0.7916 and for Urdu n-gram bleu score achieved by the system is 0.6054; METEOR score achieved is 0.8083 and F-score of 0.8250.

References

[1] Akeel M. and Mishra R., "ANN and Rule Based Method for English to Arabic Machine Translation," *The International Arab Journal of Information Technology*, vol. 11, no. 4, pp. 396-405, 2014.

[2] Banerjee S. and Lavie A., "METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments," *in Proceedings of The ACL Workshop on Intrinsic And Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, Ann Arbor, pp. 65-72, 2005.

[3] De Marneffe M., MacCartney B., and Manning C., "Generating Typed Dependency Parses from Phrase Structure Parses," *in Proceedings of the 5th International Conference on Language Resources and Evaluation*, Genoa, pp. 449-454, 2006.

[4] Hutchins W., *Machine Translation: Past, Present, Future*, Ellis HorwoodChichester, 1987.

[5] Jain A., "Parsing Complex Sentences with Structured Connectionist Networks," *Neural Computation*, vol. 3, no. 1, pp. 110-120, 1991.

[6] Johnson R., King M., and Tombe L., "EUROTRA: A Multilingual System under Development," *Computational Linguistics*, vol. 11, no. 2-3, pp. 155-169, 1985.

[7] Lewis M., Simons G., and Fennig C., *Ethnologue: Languages of the World*, SIL International, 2015.

[8] Malik M., Boitet C., and Bhattacharyya P., "Hindi Urdu Machine Transliteration Using Finite-State Transducers," *in Proceedings of the 22nd International Conference on Computational Linguistics*, Manchester, pp. 537-544, 2008.

[9] Masica C., *The Indo-Aryan Languages*, Cambridge University Press, 1993.

[10] Mishra V. and Mishra R., "ANN and Rule based Model for English to Sanskrit Machine Translation" *INFOCOMP Journal of Computer Science*, vol. 9, no. 1, pp. 80-89, 2009.

[11] Nirenburg S., *Machine Translation: Theoretical and Methodological Issues*, Cambridge University Press, 1987.

[12] Papineni K., Roukos S., Ward T., and Zhu W., "BLEU: A Method for Automatic Evaluation of Machine Translation," *in Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, Philadelphia, pp. 311-318, 2002.

[13] Pedtke T., "US Government Support and Use of Machine Translation: Current Status," *in Proceedings of MT Summit*, San Diego, pp. 3-13, 1997.

[14] Schmidt R., *Urdu, an Essential Grammar*, Psychology Press, 1999.

[15] Shahnawaz N. and Mishra R., "An English to Urdu Translation Model Based on CBR, ANN and Translation Rules," *International Journal of Advanced Intelligence Paradigms*, vol. 7, no. 1, pp. 1-22, 2015.

[16] Sinha R., "Mining Complex Predicates in Hindi Using A Parallel Hindi-English Corpus," *in Proceedings of the Workshop on Multiword Expressions: Identification, Interpretation, Disambiguation and Applications*, Singapore, pp. 40-46, 2009.

[17] Suratgar A., Tavakoli M., and Hoseinabadi A., "Modified Levenberg-Marquardt Method for Neural Networks Training," *International Journal of Computer and Information Engineering*, vol. 1, no. 6, pp. 46-48, 2005.

[18] Toma P., "Systran as A Multilingual Machine Translation System," *in Proceedings of the 3rd European Congress on Information Systems and Networks, Overcoming the Language Barrier*, München, pp. 569-581, 1977.

[19] Toutanova K., Klein D., Manning C., and Singer Y., "Feature-Rich Part-of-Speech Tagging with A Cyclic Dependency Network," *in Proceedings of Conference of The North American Chapter of The Association for Computational Linguistics on Human Language Technology*, Edmonton, pp. 173-180, 2003.

[20] Turian J., Shea L., and Melamed I., "Evaluation of Machine Translation and its Evaluation," *in Proceedings of the MT Summit IX*, New Orleans, pp. 386-393, 2003.

[21] Vikas O.; "Multilingualism for Cultural Diversity and Universal Access in Cyberspace: An Asian Perspective," *in Proceedings of Thematic Meeting for the World Summit on the Information Societ*y, UNESCO, pp. 1-49, 2005.

[22] W3Techs; Usage of Content Languages for Websites, *W3Techs.com*, Last Visited, 2015.

[23] Waibel A., Jain A., McNair A., Saito H., Hauptmann A., and Tebelskis J., "JANUS: A Speech-to-Speech Translation System Using Connectionist and Symbolic Processing Strategies," *in Proceedings of International Conference on Acoustics, Speech, and Signal Processing*, Toronto, pp. 793-796 1991.

Shahnawaz Khan (Ph. D.) is currently working as an Assistant Professor in Saudi Electronic University, Saudi Arabia. He has received his Ph. D. in Computer Science from Indian Institute of Technology, Banaras Hindu University (IT-BHU), Varanasi, India. He has around 17 years of experience in teaching and research and software industry. He is a member of editorial board in. He has been editor-in-chief for 1 international conference proceedings and 1 national conference proceedings. He is author of more than 20 research papers in refereed journals and International conferences. He has supervised 5 Master dissertations. His research interests are in smartphone sensing, natural language processing, machine learning and image processing.

Imran Usman received his BE degree in Software Engineering from Foundation University, Pakistan in 2003 and MS Computer System Engineering from Ghulam Ishaq Khan Institute of Engineering Sciences and Technology, Pakistan in 2006. He joined Pakistan Institute of Engineering and Applied Sciences as a research scholar and received his PhD degree in 2010. From 2009 to 2010 he served at Iqra University Islamabad, Pakistan as an Assistant Professor in Department of Computing and Technology. From 2010 to 2012 he served as Assistant Professor and Senior In-charge Graduate Program in the Department of Electrical Engineering at COMSATS Institute of Information Technology Islamabad, Pakistan. He is presently serving as Assistant Professor in College of Computing and Informatics, Saudi Electronic University, Kingdom of Saudi Arabia. His present research interests include machine learning, digital image processing, evolutionary computation and digital watermarking. Dr. Usman has a number of research papers to his credit and has supervised many BS, MS and PhD students. He has been awarded Research Productivity Award in 2011, 2012, 2013 and 2014 by COMSATS Institute