International Journal of Advanced Computer Research (ISSN (print): 2249-7277 ISSN (online): 2277-7970)
Volume-4 Number-2 Issue-15 June-2014

Automatic Ranking of Machine Translation Outputs Using Linguistic Factors

Pooja Gupta[1], Nisheeth Joshi[2], Iti Mathur[3]

Abstract

Machine Translation is the challenging problem in Indian languages. The main goal of MT research are to develop an MT systems that consistently provide high accuracy translations and that have broad coverage to handle the full range of languages. At an age of Internet and Globalization MT have a great demand. Since MT is an automated system; therefore, it is not necessary that the system will provide us the accurate translated output. To know the accuracy of the output, ranking of MT engines is required. There are many applications and statistical measures for computing the analysis of the performance of various MT engines based on various criteria; the oldest is by using human judges which can tell the quality of a translation, while newer automated methods include some linguistic factors. Human ranking is slow, time consuming and very tedious task. It takes too long to provide ranks for MT engine outputs. Due to this problem, a need for automatic ranking of MT outputs is required. For that we provide some automatic ranks for selecting the best translation among options from multiple systems which correlates better with humans.

Keywords

Language Modeling, Machine Translation, POS Tagging, Stemming, Quality Estimation.

1. Introduction

This research work totally depends on the result of automatic ranking of MT outputs which are independent of human intervention. MT systems are becoming widespread, embedded in more complex systems.

Manuscript received April 29, 2014.
Pooja Gupta, Department of Computer Science, Banasthali University, Rajasthan, India.
Nisheeth Joshi, Department of Computer Science, Banasthali University, Rajasthan, India.
Iti Mathur, Department of Computer Science, Banasthali University, Rajasthan, India.

There are lots of language variations, unrealistic expectations and bad translations are available in MT systems. To overcome this problem we come up with a solution, i.e. Multi-Engine Machine Translation Systems. Sometimes it also gives bad results as it cannot predict as a correct MT output. Thus, for predicting the correct MT output we require automatic ranking for a large amount of data with minimum time. Automatic ranking is generally addressed using some machine learning techniques to predict the good quality MT output.

In this paper, we proposed an approach which has used some linguistic factors. It is a fast and cheap approach and it can be done in an easy and accessible way. This approach compares the result of different Machine Translated Outputs with the human translation and check the closeness of the result. The closest result becomes the best output. In this research, we describe the results of Human Ranking using some scale based parameters as shown by Joshi et al. [1]. In this paper, we have focused on English-Hindi language pair. We have performed several tasks to accomplish the best MT output; like corpus creation, design and development of various morphological analyzers and a POS Tagger for Hindi language. The result of automatic ranking aims to help Researchers, Linguists, Language Computing Experts, Users and Software Developers of MT systems to understand as to which engine provides best translation of an English sentence. The rest of the paper is organized as follows: Brief overview of related work in presented in Section 2. In Section 3, we show how automatic ranking is performed. Here, we will explain the evaluation and the results of the research are shown in Section 4. Finally we will provide the conclusion of the study in Section 5.

2. Related Work

Statistical Machine Translation systems make use of Bayesian inference also known as Noisy Channel approach. It has a Translation Mode and a Language Model which uses an n-gram approach and refines the text in a particular language. Reordering refers to the proper positioning of text words [2]. Progress in this area is being made for several years. There are many scholars who have worked in this area and are

still working. Among them some are as follows - Specia et al. [3] have investigated the problem of predicting the quality of sentence produced by MT systems when reference translations are not available. Moreau et al. [4] have used various approaches in which several features are used to predict the quality scores. Regression algorithms are also used to predict the scores using weka toolkit. Various methods used were linear regression, space regression, support vector machines for regressions, decision trees for regression. Avramidis [5] showed an evaluation method for ranking the outputs using grammatical features. They used statistical parser to analyze and generate ranks for several MT output. Gupta el al. [6] [7] applied a Naïve bayes classifier to build model using features which are extracted from the input sentences and estimate the quality of English-Hindi outputs.

Stemming was first introduced by Lovins [8] in 1968 was proposed the use of it in Natural Language Processing applications. Porter [9] in 1980 contributed in this approach. He suggested a suffix stripping algorithm which is still considered to be a standard stemming algorithm. The proposed algorithm is one of the most accepted methods for stemming where automatic removal of affixes is done from English words. Goldsmith [10] proposed an unsupervised approach to model morphological variants of European languages. Ameta et al. [11] proposed a lightweight stemmer for Gujarati, they showed an implementation of a rule based stemmer of Gujarati and created rules for stemming and the richness in morphology. They even used it in the development of a factored machine translation system for Gujarati-Hindi language pair [12]. Paul et al. [13] developed a Hindi lemmatizer which generates rules for removing the affixes along with the addition of rules for creating a proper root word. Gupta et al. [14] developed a rule based Urdu stemmer which gave an accuracy of 86.5% as it could not perform on derivational words. Singh et al. [15] built a POS tagger for morphologically rich language in Hindi. They have achieved the best accuracy of 94.89% and an average accuracy of 94.38%. Joshi et al. [16] gave a HMM based POS tagger for Hindi. They have used IL POS tag set for the development of this tagger. They have achieved the accuracy of 92%. Shrivastava et al. [17] describes a simple HMM based POS tagger, which employs a naive (longest suffix matching) stemmer as a pre-processor to achieve reasonably good accuracy of 93.12%. Singh et al. [18] [19] proposed several POS taggers for Marathi and achieved accuracies between 77-93% for different approaches.

3. Proposed Work

Our approach tries to find the best measure to estimate the quality of MT outputs. In this paper we have used linguistic factors for ranking six MT Engine Outputs. For the purpose of automatic ranking we will use one of the most basic tasks of Machine Translation as well as Natural Language Processing such as POS-tagging and stemming. Our proposed approach is based on a trigram language model known as a baseline approach. A trigram approximation is the decomposition of the probability using the Markov assumption order 3. For example, if we want to compute the probability of a string W= $(w_1, w_2, \ldots, w_n)$ then probability estimation of a trigram on these given sentences is shown in Equations 1.

$$P(w_{n-2}w_{n-1}w_n) = \frac{Count(w_{n-2}w_{n-1}w_n)}{Count(w_{n-2}w_{n-1})} \quad (1)$$

1. Corpus Creation

We collected a corpus of 35000 sentences of English which are then translated in Hindi language by six Machine Translators. We have created our ranking system mainly for raw text of tourism domain. The approach for creation of the corpus is based on trigram language modeling. We also had a need for English-Hindi parallel lexicons, so we have used GIZA++ to generate these lexicons which have been manually checked and corrected.

a. Collection of Parallel Data

We collected a large amount of text and obtained trigrams along with their number of occurrences or frequency. We have used a total of 35000 Hindi sentences giving a total of 53062 trigram word units. Other corpora that we have created were POS-tagged trigram corpus and stemmed trigram corpus on 35000 Hindi sentences.

b. Cleaning of Corpus

We have broken the sentences and arranged them into a text file. Table1 shows an English sentence and its translated Hindi sentence. After applying a Rule based Hindi Stemmer and Hindi POS-tagger, we got stemmed and POS-tagged Hindi sentence. Stemmed and POS-tagged Hindi trigram corpus of above Hindi sentence is shown in tables 2 and 3 respectively.

Table 1: Corpus creation

English Sentence	Indians must take protective actions to protect their freedom
Hindi Sentence	भारतीयों को अपनी सवतंत्रता की रक्षा के लिये रक्षात्मक कदम उठाने चाहिए |
Stemmed Sentence	भारतीय को अपनी सवतंत्र की रक्षा के लिये रक्षा कदम उठाना चाहिए |
POS-tagged Sentence	भारतीयों/NN को/PSP अपनी/PRP सवतंत्रता/NN की/PSP रक्षा/NN के/PSP लिये/PSP रक्षात्मक/JJ कदम/NN उठाने/VM चाहिए/VAUX |/SYM

Table 2: Stemmed Corpus

S.No.	Hindi Trigrams	Stem Trigrams
1	भारतीयों को अपनी	भारतीय को अपनी
2	को अपनी सवतंत्रता	को अपनी सवतंत्र
3	अपनी सवतंत्रता की	अपनी सवतंत्र की
4	सवतंत्रता की रक्षा	सवतंत्र की रक्षा
5	की रक्षा के	की रक्षा के
6	रक्षा के लिये	रक्षा के लिये
7	के लिये रक्षात्मक	के लिये रक्षा
8	लिये रक्षात्मक कदम	लिये रक्षा कदम
9	रक्षात्मक कदम उठाने	रक्षा कदम उठाना
10	कदम उठाना चाहिए	कदम उठाना चाहिए

Table 3: POS-trigrams Corpus

S.No.	Hindi Trigrams	POS Trigrams
1	भारतीयों को अपनी	NN PSP PRP
2	को अपनी सवतंत्रता	PSP PRP NN
3	अपनी सवतंत्रता की	PRP NN PSP
4	सवतंत्रता की रक्षा	NN PSP NN
5	की रक्षा के	PSP NN PSP
6	रक्षा के लिये	NN PSP PSP
7	के लिये रक्षात्मक	PSP PSP JJ
8	लिये रक्षात्मक कदम	PSP JJ NN
9	रक्षात्मक कदम उठाने	JJ NN VM
10	कदम उठाना चाहिए	NN VM VAUX

Table 4: MT Systems

Engine No.	Description
E1	Microsoft Bing MT Engine[1]
E2	Google MT Engine[2]
E3	Babylon MT Engine[3]
E4	Moses Syntax Based Model
E5	Moses Phrase Model
E6	Example Based MT Engine

2. Machine Translators Used

For our study we have used a test corpus of 1320 English sentences and used six MT engines. This corpus was same that was used by Joshi [20] for his MT evaluation study. The MT engines that were used are listed in Table 4. First three MT engines E1, E2 and E3 are online machine translators. They are easily accessible on internet. And last three MT engines E4, E5 and E6 are developed using different MT toolkits. E4 was a MT system which used syntax based model [21] and it was trained using the Moses MT toolkit [22]. To train the system we used the Collins parser to generate parses of English sentences. E5 was a simple phrase based MT system which also used Moses MT toolkit. Joshi et al. [23] [24] had developed an example based MT system i.e. E6. These MT systems used the 35000 English-Hindi parallel corpus to train and tune themselves. We used 80-20 ratio for training and tuning of the systems i.e. we used 28000 sentences to train the systems and remaining 7000 sentences to tune the systems.

3. Methodology

In our approach, we have used the effectiveness of language models and linguistic factors in ranking MT systems. For this we had generated language models for English, Hindi as well as a Hindi Stemmed Text and also for Hindi POS Tagged Text. These LMs were already developed by Gupta et al. [25] so we have used them as it is in our study.

[1] http://www.microsofttranslator.com
[2] http://translate.goolge.com
[3] http://translation.babylon.com

a. Hindi Stemmer

Our Hindi stemmer learns suffixes automatically from a large vocabulary of words extracted from raw text. This vocabulary is known as a knowledgebase or an exhaustive lexicon list, which is created for storing the grammatical features. The working of rule based stemmer is shown in Figure1.

Here, when a user enters an input word "राष्ट्रीयता". The input word is checked in the knowledgebase. If it is present in the knowledgebase then the result is provided otherwise the word is matched with different rules created for stemming. Thus, with the help of these rules, we have reduced the word to "राष्ट्र" as the root word and "ीयता" as the suffix.

b. Hindi POS tagger

Part-of-speech tagging is assigning the words in a text as corresponding to a particular part of speech. We have used a POS tagger for Hindi language developed by Joshi et al. [16] and made some modifications on it. This system was augmented by adding some rules to bypass un-necessary processing. In rule base, we applied a set of hand written rules and contextual information to assign POS tags to words. Then, on the remaining words, we applied HMM POS tagger that assigned the best tag to a word by calculating the forward and backward probabilities of tags along with the sequences provided as an input. For calculating backward and forward tag probabilities we use equation 2.

$$P(t_i|w_i) = P(t_i|t_{i-1}).P(t_{i+1}|t_i).P(w_i|t_i) \quad (2)$$

We have defined the context of the tags (backward and forward) with respect to the current tag using HMM. We performed this operation for each word in the corpus. This context phenomenon is a very powerful feature of HMM POS tagger which can decide the tag for a word by looking at the tag of the previous word and the tag of the future word. For developing a POS tagger we first required to annotate a corpus based on a tag set. We used the IL POS tag set [12]. After assigning the tags on MT outputs, we can apply ranking algorithm and get the best MT output.

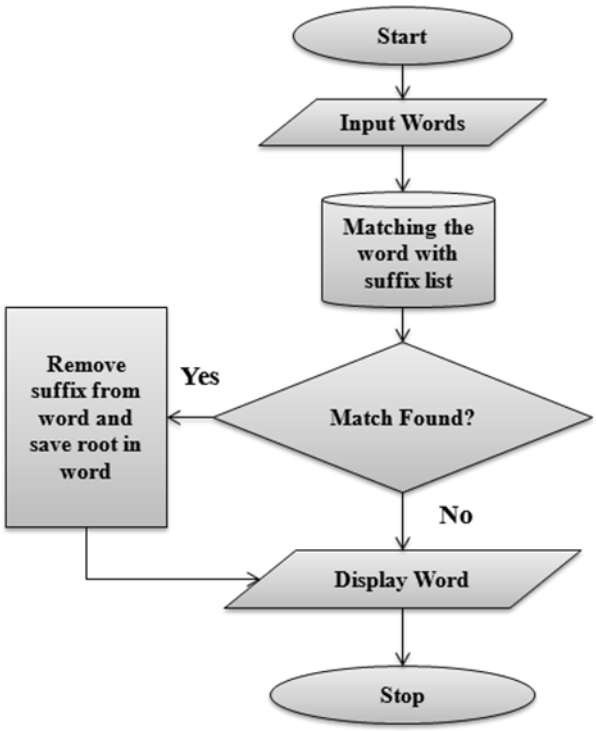

Figure 1: Stemming System

c. Ranking System

We have generated language models for English, Hindi as well as a Hindi Stemmed Text and also for Hindi POS Tagged Text. Along with English sentence and MT outputs, we also provided stemmed MT outputs and POS Tagged MT Outputs. Then we applied the ranking algorithm to rank these six MT engine outputs and get ranked MT output list.

Ranking Algorithm

Step1. Trigrams from stem and POS tagged sentences are generated separately.
Step2. These trigrams are matched with stem and POS tagged language model separately and matched ones are retained.
Step3. Match retained Hindi stemmed trigram's lexicons and POS tagged trigram's lexicons with the Hindi lexicon list.
Step4. If a match is found then register corresponding Hindi stem lexicon and the Hindi POS tagged lexicon.
Step5. Match Hindi language model with registered Hindi stem lexicons as well as Hindi POS tagged lexicons and sum the probabilities of each match.
Step6. Compute the average of all these probabilities.
Step7. Perform these steps on all MT outputs.

Step8. Sort these average probabilities of MT outputs in descending order with respect to their cumulative probabilities.

We have illustrated the entire ranking process through the following example to have a better understanding of the functionality of ranking system.

Sentence: India is a vast country known for its diversified culture and traditions.

E1 Output: भारत एक विशाल देश अपने विविध संस्कृति और परंपराओं के लिए जाना जाता है।

E2 Output: भारत एक विशाल देश अपने विविध संस्कृति और परंपराओं के लिए जाना जाता है.

E3 Output: भारत एक विशाल देश के लिए उसके नाम से प्रसिद्ध विविधीकृत संस्कृति और परंपराएं हैं।

E4 Output: भारत का एक नांदी देश के लिए अपने diversified संस्कृति और traditions.

E5 Output: India एक vast देश के लिए जाना जाता है इसकी diversified culture और traditions है।

E6 Output: भारत देश के लिए जाना जाता है एक विस्तृत अपनी संस्कृति और परम्पराओं ।

Table 5 shows the trigram statistics of these sentences and also shows the cumulative probabilities and its average probabilities of these trigrams. Finally we apply Step 8 of ranking algorithm and we can rank the system according to their average probabilities.

Table 5: MT Systems

Engine	Trigrams	Prob. Sum	Prob. Average	Ranked Output
E1	12	10.2948	3.43162	1
E2	12	10.0953	3.36511	2
E3	13	5.6060	1.86868	4
E4	10	3.2993	1.09979	5
E5	13	6.6850	2.22835	3
E6	11	2.5641	0.85473	6

4. Evaluation

a. Evaluation of Hindi Stemmer

To evaluate the Hindi rule based stemmer system we used the approach used by Paul et al. [26]. Since, we wanted to know the accuracy of the system. We used the following formula:

$$\text{Accuracy}\ (\%) = \frac{\text{Accurate stemmed}}{\text{Total no of given word}} \times 100$$

Here we checked our system on the test data of 5000 sentences and total 112345 words out of which 90104 words gave correct stem. By using the above formula, we achieved the accuracy of 80.20%. Figure 2 shows the result of this evaluation.

b. Evaluation of Hindi POS Tagger

To evaluate the Hindi POS tagger, we developed a POS-tagged corpus of 1300 Hindi sentences. To evaluate the system we used the same measure as that was used by Singh et al. [27]. They used Precision, Recall and F-Measure to calculate the accuracy of the system and were calculated using the following formula.

$$\text{Precision}\ (P) = \frac{\text{No. of correct POS tags assigned by the system}}{\text{No. of POS tags assigned by the system}}$$

$$\text{Recall}\ (R) = \frac{\text{No. of correct POS tags assigned by the system}}{\text{No. of POS tags in the text}}$$

$$F - \text{Measure} = \frac{2 \times P \times R}{P + R}$$

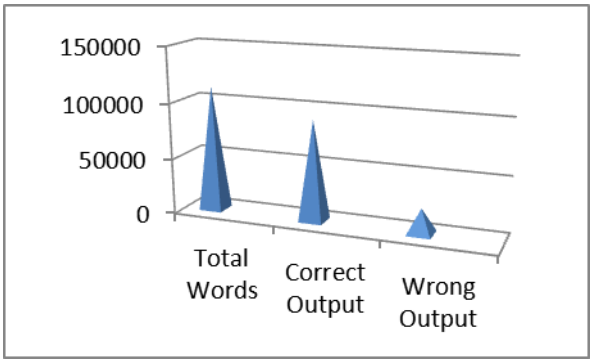

Figure 2: Result of Test data

Test scores of our system are as follows:

No. of Correct POS tags assigned by the system = 20849

No. of POS tags assigned by the system = 19364

No. of POS tags in the text = 19364

Thus accuracy of the POS tagger system is 92.87%.

Table 6: Ranking Evaluation Scale

Score	Description
1	Excellent
2	Good
3	Average
4	Poor
5	Bad

Table 7: Ranking at Combined Category

Engine	Stem POS LM Ranking	Baseline Ranking	Human Ranking	Evaluation Rank
E1	407	467	451	Excellent
E2	285	290	279	Good
E3	145	64	140	Poor
E4	8	77	22	Poor
E5	256	186	205	Bad
E6	236	223	240	Average

Table 8: Ranking at Web-Based Category

Engine	Stem POS LM Ranking	Baseline Ranking	Human Ranking	Evaluation Rank
E1	633	663	669	Excellent
E2	462	439	498	Good
E3	242	235	170	Poor

Table 9: Ranking at MT Toolkits Category

Engine	Stem POS LM Ranking	Baseline Ranking	Human Ranking	Evaluation Rank
E4	116	141	125	Bad
E5	634	471	497	Poor
E6	756	725	715	Excellent

c. Evaluation of Ranking System

To evaluate the performance of the overall ranking system we used 1320 English sentences from tourism domain. We collected the translations of six machine translators. Then we collected stems and POS tags of these 1320 Hindi sentences. These sentences were not part of our 35000 sentences that were used to train the models. To validate our results we compared the ranks of our system with the ranks given to MT systems by a human evaluator. Human evaluator used a subjective human evaluation that was used by Gupta et al. [28] [29]. The evaluation of an MT output was done on the basis of ten parameters. These were shown by Joshi et al. [30]. Each MT outputs were adjudged on these 10 parameters.

We evaluated the system generated ranks with baseline system ranks and human ranks in three different categories. In the first category we compared the ranks of all these systems, irrespective of their type. This category is known as combined category. In the second category we compared the ranks of only web based systems. In third category we compared the ranks of only MT toolkits or systems. The human ranking, an evaluator was asked to give a score on a 5-point scale as shown in Table 6. Table 7, 8 and 9 shows the results of the combined category; Web based category and MT Toolkits category respectively. Figure 3, 4 and 5 summarize these data.

5. Conclusion

In this research work, we have introduced an approach for providing ranks on six machine translation engine outputs. For this, we have used 1320 sentences for testing the systems which are from tourism domain. We have generated trigram language models for Hindi stemmed text as well as Hindi POS tagged text. The system described here are very simple and efficient for automatic ranking even when the amount of available raw text is large. We can show that by using linguistic factor based ranking, the accuracy of the systems fall below as that of the baseline model. If we compared the results of linguistic based LM ranking with human ranking then the results are comparable. Moreover, we can clearly see that a simple phrase based SMT system which was termed as a poor performer by the human judges got a good score with baseline ranking but was adjudged as not so good by linguistic factor-based ranking.

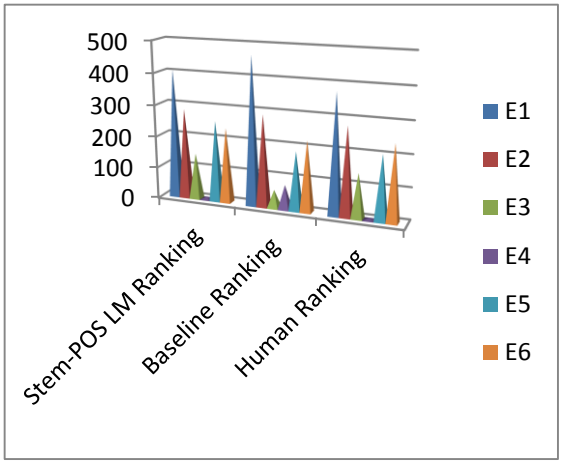

Figure 3: Ranking at Combined Category

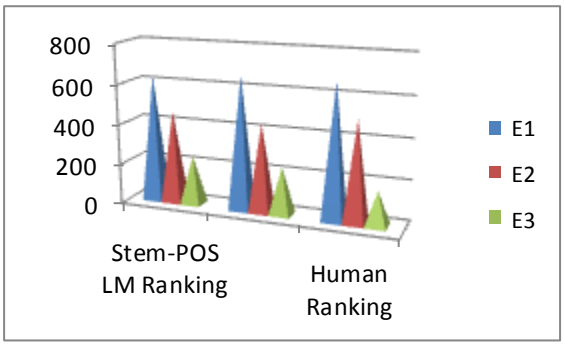

Figure 4: Ranking at Web-Based Category

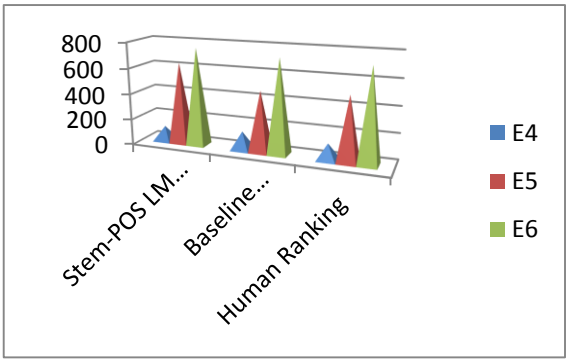

Figure 5: Ranking at MT Toolkits Category

References

[1] N. Joshi, I. Mathur, H. Darbari, and A. Kumar, "HEval: Yet Another Human Evaluation Metric." International Journal of Natural Language Computing, Vol 2, No 5, pp 21-36. 2013.

[2] P. Koehn, "Statistical Machine Translation", Cambridge University Press, pp 127-130, 314-319. 2009.

[3] L. Specia, M. Turchi, N. Cancedda, M. Dymetman, and N. Cristianini, "Estimating the Sentence-Level Quality of Machine Translation Systems". In 13th Annual Meeting of the European Association for Machine Translation (EAMT-2009), pages pp. 28–35, Barcelona, Spain. 2013.

[4] E. Moreau, and C. Vogel, "Quality estimation: an experimental study using unsupervised similarity measures". In Proceedings of the Seventh Workshop on Statistical Machine Translation, pp. 120-126. Association for Computational Linguistics. 2012.

[5] E. Avramidis, "Quality Estimation for Machine Translation output using linguistic analysis and decoding features". In Proceedings of the 7th Workshop on Statistical Machine Translation, Montre´al, Canada June7-8, 2012

[6] R. Gupta, N. Joshi, I. Mathur, "Analysing Quality of English-Hindi Machine Translation Engine Outputs Using Bayesian Classification." International Journal of Artificial Intelligence and Applications, Vol 4 (4), pp 165-171. 2013.

[7] R. Gupta, N. Joshi, and I. Mathur. "Quality Estimation of English-Hindi Outputs Using Naïve Bayes Classifier." Advances in Computing, Communications and Informatics (ICACCI), 2013 International Conference on. IEEE, 2013.

[8] J. B. Lovins, "Development of Stemming Algorithm", MIT Information Processing Group, Electronic Systems Laboratory, 1968.

[9] M. F. Porter, "An algorithm for suffix stripping". Program: electronic library and information systems 14(3), pp 130-137. 1980.

[10] J. Goldsmith, "An algorithm for unsupervised learning of morphology", Natural Language Engineering. 12(4), pp 353-371. 2006.

[11] J. Ameta, N. Joshi, I. Mathur, "A Lightweight Stemmer for Gujarati." In Proceedings of 46th Annual National Convention of Computer Society of India. Ahmedabad, India, 2011.

[12] J. Ameta, N. Joshi, I. Mathur, "Improving the Quality of Gujarati-Hindi Machine Translation Through Part-of-Speech Tagging and Stemmer-Assisted Transliteration." International Journal on Natural Language Computing, Vol 3(2), pp 49-54, 2013.

[13] S. Paul, N. Joshi, I. Mahtur, "Development of a Hindi Lemmatizer." International Journal of Computational Linguistics and Natural Language Processing, Vol 2(5), pp 380-384, 2013.

[14] V. Gupta, N. Joshi, I. Mathur, "Rule Based Urdu Stemmer." In Proceedings of 4th International Conference on Computer and Communication Technology. IEEE, 2013.

[15] S. Singh, et al., "Morphological richness offsets resource demand-experiences in constructing a POS tagger for Hindi." Proceedings of the COLING/ACL on Main conference poster sessions. Association for Computational Linguistics, 2006.

[16] N. Joshi, H. Darbari and I. Mathur, "HMM Based POS Tagger for Hindi", In Proceedings of International Conference on Artificial Intelligence, Soft Computing. 2012.

[17] M. Shrivastava and P. Bhattacharyya, "Hindi POS Tagger Using Naïve Stemming: Harnessing Morphological Information without Extensive Linguistic Knowledge." International Conference on NLP (ICON08), Pune, India, 2008.

[18] J. Singh, N. Joshi, and I. Mathur, "Part of Speech Tagging of Marathi Text Using Trigram Method", International Journal of Advanced

Information Technology, pp 35-41, Vol 3. No. 2. 2013.

[19] J. Singh, N. Joshi, and I. Mathur, "Development of Marathi Part of Speech Tagger Using Statistical Appraoch." Advances in Computing, Communications and Informatics (ICACCI), 2013 International Conference on. IEEE, 2013.

[20] N. Joshi, "Implications of linguistic feature based evaluation in improving machine translation quality a case of english to hindi machine translation." 2014.

[21] H. Hoang, and P. Koehn. "Improved translation with source syntax labels." Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR. Association for Computational Linguistics, 2010.

[22] Koehn et al., "Moses: Open source toolkit for statistical machine translation." In Proceedings of the Annual Meeting of the Association for Computational Linguistics, demonstration session. 2007.

[23] N. Joshi, I. Mathur, and S. Mathur, "Translation Memory for Indian Languages: An Aid for Human Translators." Proceedings of 2nd International Conference and Workshop in Emerging Trends in Technology. 2010.

[24] N. Joshi, and I. Mathur, "Design of English-Hindi Translation Memory for Efficient Translation." In Proc. of National Conference on Recent Advances in Computer Engineering.2012.

[25] P. Gupta, N. Joshi, and I. Mathur, "Automatic Ranking of MT Outputs using Approximations." International Journal of Computer Application, Vol 81(17), pp 27-31. 2013.

[26] S. Paul, M. Tandon, N. Joshi, I. Mahtur, "Design of a Rule Based Hindi Lemmatizer". In Proceedings of Third International Workshop on Artificial Intelligence, Soft Computing and Applications, Chennai, India, pp 67-74, 2013.

[27] J. Singh, N. Joshi and I. Mathur, "Marathi Part-of-Speech Tagger Using Supervised Learning." Intelligent Computing, Networking, and Informatics. Springer India, 2014. 251-257.

[28] V. Gupta, N. Joshi, I. Mathur, "Evaluation of English-to-Urdu Machine Translation." Intelligent Computing, Networking, and Informatics. Springer India, 2014. 351-358.

[29] V. Gupta, N. Joshi, I. Mathur, "Subjective and Objective Evaluation of English to Urdu Machine Translation." Advances in Computing, Communications and Informatics (ICACCI), 2013 International Conference on. IEEE, 2013.

[30] N. Joshi, H. Darbari, I. Mathur, "Human and Automatic Evaluation of English to Hindi Machine Translation Systems." Advances in Computer Science, Engineering & Applications. Springer Berlin Heidelberg, 2012. 423-432.

Pooja Gupta has completed her M.Tech in Computer Science from Banasthali University, Rajasthan. She is a Research Scholar in English-Indian Languages Machine Translation System Project sponsored by TDIL Programme, DeitY. Her current research interest includes Natural Language Processing and Machine Translation. Her research paper entitled "Automatic Ranking of MT engine Outputs using Approximations" was published by International Journal of Computer Applications, 81(17), 27-31, November 2013.

Dr. Nisheeth Joshi is an Associate Professor at Banasthali University. He has been primarily working in design and development of evaluation Metrics in Indian languages. Besides this he is also actively involved in the development of MT engines for English to Indian Languages. He is one of the experts empanelled with TDIL programme, Department of Electronics and Information Technology (DeitY), Govt. of India, a premier organization which foresees Language Technology Funding and Research in India. He has several publications in various journals and conferences and also serves on the Programme Committees and Editorial Boards of several conferences and journals.

Iti Mathur is an Assistant Professor at Banasthali University. Her primary area of research is Computational Semantics and Ontological Engineering. She is also a Co-Principal Investigator of English to Indian Language Machine Translation Development System Funded by Govt. of India. The project is a consortium mode project, where 13 institutions are developing machine translators from English to 8 different Indian languages. She has several publications in various journals and conferences and also serves on the Programme Committees and Editorial Boards of several conferences and journals.