Sifar: An Attempt to Develop Interactive Machine Translation System for English to Hindi

Meenal Jain, Mehvish Syed, Nidhi Sharma, Shambhavi Seth and Nisheeth Joshi

Abstract The presently available machine translation systems are still far from being perfect, and to improve their performance the concept of interactive machine translation (IMT) was introduced. This paper proposes Sifar, an IMT system, which uses statistical machine translation and a bilingual corpus on which several algorithms (Word error rate, Position Independent Error Rate, Translation Error Rate, n-grams) are implemented to translate text from English to Hindi. The proposed system improves both the speed and productivity of the human translators as found through experiments.

Keywords Machine translation · Statistical machine translation · Computer aided translation · Interactive machine translation · Position independent error rate · Word error rate · Translation error rate

1 Introduction

Since the very advent of languages, translations emerged as a very important aspect to minimize the communication gap. With time and computational capacity, a vast number of natural languages are being digitized. In India alone, we have such large

M. Jain · M. Syed · N. Sharma · S. Seth (✉) · N. Joshi
Department of Computer Science, Banasthali Vidyapith, Vanasthali, Rajasthan, India
e-mail: shambhavi22seth@gmail.com

M. Jain
e-mail: mpbjain@gmail.com

M. Syed
e-mail: mehvishsyed97@gmail.com

N. Sharma
e-mail: vats.nidhi10@gmail.com

N. Joshi
e-mail: jnisheeth@banasthali.in

A. K. Luhach et al. (eds.), *First International Conference on Sustainable Technologies for Computational Intelligence*, Advances in Intelligent Systems and Computing 1045,
https://doi.org/10.1007/978-981-15-0029-9_54

multilingual diversity. It becomes a necessity to have translators for efficient communication and having human translators was tedious and both resource and time-consuming. This gave birth to machine translation systems. In such a fast-paced world where time plays a significant role, MT requires certain enhancements which were introduced as CAT (computer-aided translation) tools. The current state-of-the-art MT systems do not produce ready-to-use translations; thus, human post-editing is required to achieve high-quality translations. To address this issue, the MT systems are augmented with human translators to give birth to a new technology, namely interactive machine translation which provides a framework for adaptive learning to help the user with sentence and phrase completion suggestions during the translation process. Many IMT systems have so far been developed for foreign languages, but negligible efforts have been made for Indian languages. So, we thought of developing a tool that could allow an English speaker to translate in Hindi and developed translation systems lack good quality in open domain automatic translation. Since they do not have an open domain ability to translate text, we have to move towards a translation technique where a first-hand translation can be co-headed by machine and then can be perfected by a human. Hence, Sifar is an attempt to develop a system for English to languages translation, and thus it is an IMT so that user can get the translation with multiple suggestions. Sifar aims at assisting human translators to accelerate the overall translation process with the help of an example database. As the user translates the text, the translations are added to the example database, and when a similar sentence reoccurs, the previous translation is inserted into the translated document, in turn saving the user's effort of re-translating that sentence, and is particularly useful when translating a revision of a previously translated text.

Section 2 describes the related work done in this area. Section 3 discusses our methodology where we have shown the development process of Sifar. Section 4 evaluates the system, and Sect. 5 concludes the same.

2 Related Work

Nagao [10] proposed the idea of machine translation based on the examples, which were made available to the system prior to the translation process. A bilingual corpus was used which was comprised of source language text and their corresponding target language text. This approach was based on how humans process a sentence in source language and translate it to target language. Machine translation system use natural languages, which are highly complex (use of homonyms, different grammatical rules), to make decisions. In RBMT, defining rules is a difficult task and cost of building dictionaries and making changes is high. To overcome the problem of knowledge acquisition, Corpus-based MT comprising of SMT and EBMT was used. Languages with varying word orders proved to be a challenge for SMT. However, for the production of dependency trees for sentence analysis and example database, EBMT requires analysis and generation modules. Hybrid approach minimizes the challenges of other approaches [11]. Block transpositions [9] are used to extend edit distance to define inversion edit distance as a metric of the cost of parsing for a sentence pair within an inversion grammar. Experiments showed that at system level, correlation of automatic evaluation with human judgment is appropriate. DerivTool [2], an interactive translation visualization tool provides with a myriad number of options for the user to choose from and allows her to look into the decoding process where syntax-based framework for translation has been used. Moses [5], an open source toolkit for SMT, featured the phrase-based translation with factors and confusion network decoding, which allowed translation of ambiguous input, along with the capabilities of Pharaoh decoder [6]. Caitra, developed by Koehn et al. [7] based on the TransType Project [8], a web-based IMT tool, with the help of Moses decoder, featured making suggestions for sentence completion, alternative word provision and phrase translation, and giving post editing options with key stroke logging for detailed analysis. Cunei [12] proposed a hybrid system comprising of features of both EBMT and SMT, by using the example base to model each phrase pair at run time and perform recombination to get the translations. The system was tested using three language pairs Finnish to English, French to English, and German to English against the Moses model, where Cunei displayed unexpectedly better results for German to English. Due to German compounding, it becomes difficult for the one to many alignment and phrase extraction, but Cunei had advantage due to its runtime modeling as it adapts itself over time by updating weights. The complex lexicalized distortion model gave nearly the same result for the reordering model which was informed only about how far and frequently the phrases were moved during decoding. CASMACAT [1], a modular, web-based translation workbench with advanced functionality and editing features provided us with TM for raw match and automatic translations from MT server for post editing, consisting of a GUI, backend, an MT server, and a CAT server. The main features included interactive translation prediction (ITP), confidence measures, prediction length control, search and replace, word alignment information, one clicks rejection, replay mode and logging function, and e-pen for handwritten interaction. The system was evaluated at Celer Soluciones, where 9 professors participated to carry out post-editing for English–French translations. Three workbench features were tested: post-editing, intelligent autocompletion (IA), and ITP where ITP had a slower rate due to the unfamiliarity with the IA system. Also due to the lack of visual aid to control IA, double checking had to be done even for small post-editing tasks. Dungarwal et al. [3] proposed a Hindi–English Translation consisting of phrase based and factored statistical machine translation (SMT) systems that considered number, case, and tree adjusting grammar information as factors. Translation of out of vocabulary and transliteration of named entity are done at the preprocessing stage. Merging the chunks, preposition chunk reordering, verb chunk reordering was used as the rules of development. Wang et al. [14] proposed CytonMT, an open source toolkit which emphasize on neural machine translation. The system use attention-based RNN encoder-decoder and is developed on C++. It uses NVIDIA GPU accelerated libraries. The proposed system proved to be efficient on BLUE score and provides a considerable speed-up in training phase of the system. Helcl et al. [4] presented a CUNI system which supports multimodal machine translation (MMT) task and incorporates self-attentive neural network instead of a recurrent neural network which proves to be an effective measure. The proposed system works on doubly attention transformer architecture and imagination concept, formerly introduced for sequence to sequence RNN model.

3 Methodology

To implement Sifar, we integrated out translator with an existing machine translation system, which has been trained statistically and is based on Moses machine translation toolkit. In order to train this statistical system, we used a mix-domain corpus which comprises of translations from tourism, health, agriculture and administrative domains. We preferred mix-domain training to achieve open domain translations. Another important factor we incorporated in our system is a bilingual corpus on which our translator is trained. The chunks in the corpus are taken from the tourism domain. When source text is provided to the system, it first looks for the sentences in the corpus, evaluates the sentences one at time, to check for their similarity with the input text provided. In case, the similarity score is greater than the set threshold value, the target text corresponding to that source text as well as the output from SMT system is joined and the first-order output is proposed to the user. As we claim our system to be interactive, we allow the user to make changes as required in the target text of most suitable candidate target text and further, add this new source–target pair as a new tuple in the corpus so that the forthcoming translations can make a use of it. Sifar calculates the similarity score of hypotheses (translation produced by system) and reference sentence (example translation available in knowledge base) using various algorithms. These algorithms are named as Word Error Rate, Position Independent Error Rate, Translation Error Rate, n-grams similarity, and all of these are discussed further.

3.1 Word Error Rate (WER)

Word error rate [13] is defined as the minimum number of edits (insertion, deletion, and substitution) required to transform the hypothesis string into reference string. It is calculated as the Levenshtein's Edit distance between the hypothesis string and the reference string, normalized by the length of reference string.

Algorithm-1 to calculate WER

Input: REFERENCES R, HYPOTHESIS h.
for all *r* in *R* **do**
  h′ h
  e′ 0
  e′ min_edit_distance(*h′; r*)/*length*(*r*)
end for
return *e*

Example
Hypothesis: The best time to visit Jaipur city is around October to March.
Result: जयपुर शहर घूमने के लिए अक्टूबर से मार्च तक तक का समय उत्तम है। 50%. जयपुर संगमरमर की मूर्तियों, नीले मिट्टी के बर्तन और राजस्थानी जूतियों के लिए भी प्रसिद्ध है। 32%.

3.2 Position Independent Error Rate (PER)

Position independent error rate [11] is similar to WER except the fact that PER does not take word order into consideration. PER counts the number of non-similar words between the hypothesis and the reference string as the number of substitutions required. The number of insertions or deletions required depends upon the difference between the length of the reference and hypothesis string. The total error rate is the sum of insertions, deletions, and substitutions normalized by the length of the reference string.

Algorithm-2 to calculate PER

Input: REFERENCES R, HYPOTHESIS h
for all *r* in *R* **do**
  h′ h
  e 0
  for all *words;words_hypothesis*in *h* **do**
    e 0
    if *word - hypothesis* is present in *r* **then**
      e e+ 1
    end if
  end for
end for
max_length max
e<-(*max_length*□ *e*)/ length of reference string
return *e*

Example
Hypothesis: The best time to visit Jaipur city is around October to March.
Result: जयपुर का बाज़ार चमकीला; और दुकानें रंग बिरंगी वस्तुओं, जिनमें हस्तकला की वस्तुएँ, बहुमूल्य पत्थरों मीनाकारी की वस्तुओं गहनों व राजस्थानी चित्रकला आदि से भरे हैं। 78.788 %
नापो, गौलेरास, कुटुक और कॉडोर की शृंखलाएँ यहाँ स्थित हैं। 42.851%.

3.3 Translation Error Rate (TER)

Translation error rate is calculated as the minimum number of edits (insertion, deletion, substitution) required on individual words as well as sequence of words, normalized by the length of reference string, to transform hypothesis string into reference string.

Algorithm-3 to calculate TER

Input: REFERENCES R, HYPOTHESIS h
for all *r* in *R* **do**
  h′ h
  e 0
  repeat
    if s reduces edit distance **then**
      h apply s to h'
      e e+ 1
    end if
  until No shifts that reduce edit distance
  e e+ *min_edit_distance*(*h′; r*)
end for
return e

Example
Hypothesis: The best time to visit Jaipur city is around October to March.
Result: जयपुर शहर घूमने के लिए ओक्टोबर से मार्च तक का समय उत्तम है। 75%.
अक्तूबर से फरवरी जैसलमर भ्रमण का श्रेष्ठ समय माना जाता है। 50%.

3.4 N-Grams

The n-gram algorithm divides the reference and the hypothesis string into chunks of n words each and then compares the chunks of the hypothesis string with those of the reference string. Sifar uses n-grams algorithm as 3-grams and 4-grams.

Algorithm-4 for N-Grams

Input: REFERENCES R, HYPOTHESIS h
h′ h
hyp[] divide h' into substrings of length n each
for all *r* in *R* **do**
    ref[] divide r into substrings of length n each
    for all *t* in *hyp*[] **do**
        q <-0
        repeat
            if *t* **equals** ref[q] **then**
                cost+ 1
                q+ 1
            end if
        until *q* **equals** length(hyp[ ])
        result cost / max(length(hyp[]), length(ref[]))
    end for
end for
return *result*

Example
Hypothesis: The best time to visit Jaipur city is around October to March.
Result: जयपुर शहर घूमने के लिए अक्तूबर से मार्च तक का समय उत्तम है। 69.230%.

Our IMT system (Sifar) made use of these algorithms to get the example translations and ranked them according to similarity. It also took the translation of an MT system and adds to the list of possible translations. These all are provided to the human translation who can select the one which the human translator feels most appropriate and perform post editing to generate a final high-quality translation. The working of this entire system is shown in Fig. 1 (Fig. 2).

4 Evaluation

We performed our evaluation on 5 human translators who were asked to translate 1000 sentences using Sifar. For the first 500 sentences, we did not provide them with any suggestive translations and where are to do translations on their own. For the rest 500 sentences, we provided the translators with suggestions. They analyzed the results based on speed and productivity. The results are as follows.

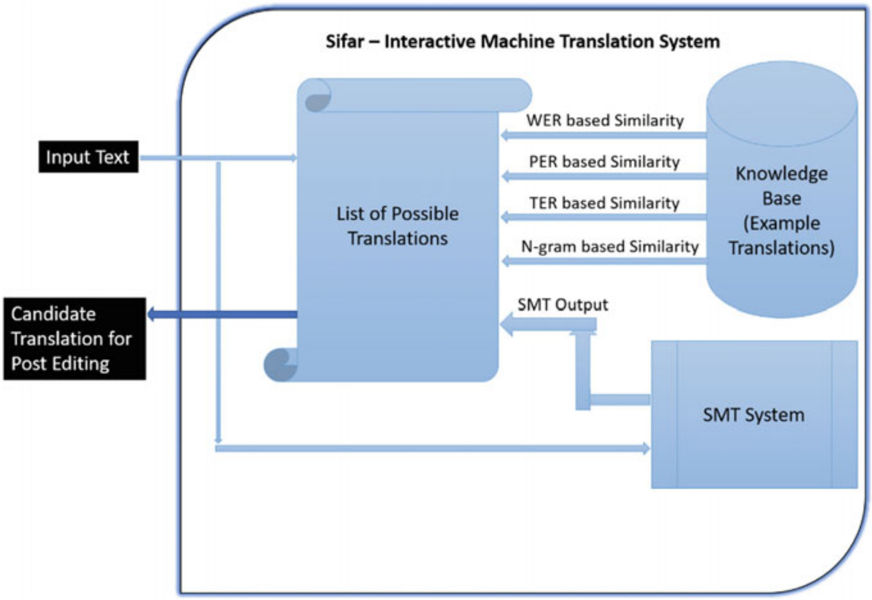

Fig. 1 Working of Sifar-interactive machine translation system

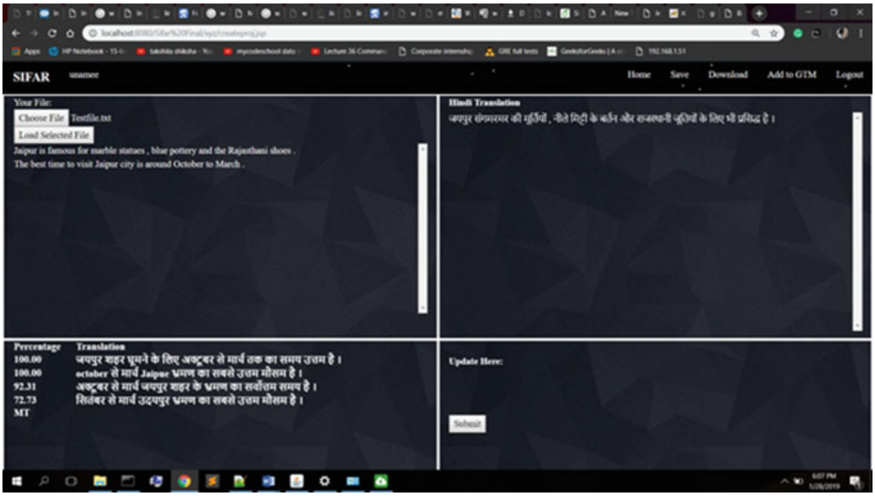

Fig. 2 Screenshot: Sifar-interative machine translation system

Table 1 Time taken to complete the translations

Human translations	Without suggestive translators (s/word)	With suggestive translations (s/word)	Difference (s/word)
H1	6.3	3.2	3.1
H2	5.3	3.4	1.9
H3	3.3	2.3	1
H4	6.5	2.6	3.9
H5	6.1	3.5	2.6

4.1 Speed

All the five translators performed very well, when they were provided with the suggestive list of possible translations. This improved their speed to doing translations. Table 1 shows the results of this experiment. In this, we calculated the speed as well as the total time taken to complete the task divided by total no. of words in the documents. This is shown in Eq. (1). Here the calculated speed of doing translations when no suggestions were provided and when the suggestions were provided.

$$\text{Speed} = \frac{\text{Total Time Taken}}{\text{Total No. of Words}} \tag{1}$$

4.2 Productivity

In order to ascertain productivity, we calculated the score of HTER (human translation edit rate). In this, we calculated the total number of edits (inserts, deletes, substitutes, sifts) performed by human translator to correct the translation which was selected from the list of suggestive translations. This was done on only last 500 sentences, as these were the sentences on which suggestive translations were provided. In all the cases, the edit rate of the translation was very less. This confirms that the use of Sifar improves the productivity of the human translators. Table 2 shows the results of this study.

Table 2 HTER score

Human translators	HTER score
H1	0.3452
H2	0.1542
H3	0.1453
H4	0.3421
H5	0.2568

5 Conclusion and Future Work

In this paper, we have shown the design of an interactive machine translation system which combines the strengths of both human and machine and which provides a high-quality machine assisted translation. Through experiments, we have verified our claim, as with the help of our system the speed of the human translators also improved. It also improved the productivity of the translators. As an extension to this study, we wish to improve the user friendliness of the system by providing several features like providing a human translator an option to add their example translations in the knowledge base, before doing actual translations. We also wish to perform a more thorough evaluation to understand the expectations of the human translators from this system. One of the possible expectations is to reduce the key in effort. We shall perform the usability evaluation of the system to analyze such expectations and incorporate the same in subsequent versions of Sifar.

We plan to improve our translation system by implementing certain modules. The first module would include expanding the translation system for other Indian languages as our current system works only on English to Hindi translations. This would increase the scope and utility of our system. The second module would incorporate the concept of neural networks to the present system. Neural machine translation systems produce better outputs as they understand and establish the similarities between the words and produce more fluent outputs. Hence, if used with our system, it would increase our system's efficiency.

References

1. Alabau, V., Buck, C., Carl, M., Casacuberta, F., García-Martínez, M., Germann, U., Mesa-Lao, B.: Casmacat: a computer-assisted translation workbench. In: Proceedings of the Demonstrations at the 14th Conference of the European Chapter of the Association for Computational Linguistics, pp. 25–28 (2014)
2. DeNeefe, S., Knight, K., Chan, H. H.: Interactively exploring a machine translation model. In: Proceedings of the ACL 2005 on Interactive poster and demonstration sessions. Association for Computational Linguistics, pp. 97–100 (2005)
3. Dungarwal, P., Chatterjee, R., Mishra, A., Kunchukuttan, A., Shah, R., Bhattacharyya, P.: The iitbombayhindi-english translation system at wmt 2014. In: Proceedings of the Ninth Workshop on Statistical Machine Translation, pp. 90–96 (2014)
4. Helcl, J., Libovický, J., Variš, D.: CUNI System for the WMT18 Multimodal Translation Task. arXiv preprint arXiv:1811.04697 (2018)
5. Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C.: Moses: open source toolkit for statistical machine translation. In: Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions. Association for Computational Linguistics, pp. 177–180 (2007)
6. Koehn, P.: Pharaoh: a beam search decoder for phrase-based statistical machine translation models. In: Conference of the Association for Machine Translation in the Americas, pp. 115–124. Springer, Berlin, Heidelberg (2004)
7. Koehn, P.: A web-based interactive computer aided translation tool. In: Proceedings of the ACL-IJCNLP 2009 Software Demonstrations, pp. 17–20 (2009)

8. Langlais, P., Foster, G., Lapalme, G.: TransType: a computer-aided translation typing system. In: Proceedings of the 2000 NAACL-ANLP Workshop on Embedded machine translation systems, vol. 5. Association for Computational Linguistics, pp. 46–51 (2000)
9. Leusch, G., Ueffing, N., Ney, H.: A novel string-to-string distance measure with applications to machine translation evaluation. In: Proceedings of Mt Summit IX, pp. 240–247 (2003)
10. Nagao, M.: A framework of a mechanical translation between Japanese and English by analogy principle. In: Artificial and Human Intelligence, pp. 351–354 (1984)
11. Okpor, M.D.: Machine translation approaches: issues and challenges. Int. J. Comput. Sci. Issues (IJCSI) **11**(5), 159 (2014)
12. Phillips, A.B., Brown, R.D.: Cunei machine translation platform: system description. In: 3rd Workshop on Example-Based Machine Translation (2009)
13. Tillmann, C., Vogel, S., Ney, H., Zubiaga, A., Sawaf, H.: Accelerated DP based search for statistical translation. In: Fifth European Conference on Speech Communication and Technology, pp. 2667–2670 (1997)
14. Wang, X., Utiyama, M., Sumita, E.: CytonMT: An Efficient Neural Machine Translation Open-source Toolkit Implemented in C++. arXiv preprint arXiv:1802.07170 (2018)