Chapter 35
A Collaborative Platform to Collect Data for Developing Machine Translation Systems

Md. Arid Hasan, Firoj Alam and Sheak Rashed Haider Noori

1 Introduction

The significant progress of Machine Translation (MT) system has started back in the late 1980s and early 1990s due to the availability of computational power and parallel corpora. Notable research work include IBM statistical machine translation models [5], phrase-based MT models by Koehn et al. [14], and hierarchical phrase-based statistical MT [6] among others. The development of open-source Statistical Machine Translation (SMT) toolkit Moses [15] facilitated the SMT research one step further. Due to the advancement of computational power and the notable work of deep learning, in the past few years, machine translation systems are moving forward with a new approach—called Neural Machine Translation (NMT) [7].

From SMT approach to NMT approach, a common idea is to learn the parameters from a large amount of parallel corpus. It is clear that these techniques are highly dependent on the human translated aligned parallel corpus. One of the publicly available and a large corpus is Europerl [13], which has been extracted from the proceedings of the European Parliament consists of 21 European languages. Developing such a resource is difficult and challenging for under-resourced languages such as Bangla.

Md. A. Hasan (✉) · S. R. H. Noori
Daffodil International University, Dhaka, Bangladesh
e-mail: arid15-5332@diu.edu.bd

S. R. H. Noori
e-mail: drnoori@daffodilvarsity.edu.bd

F. Alam
QCRI, Al Rayyan, Qatar
e-mail: fialam@hbku.edu.qa

M. S. Uddin and J. C. Bansal (eds.), *Proceedings of International Joint Conference on Computational Intelligence*, Algorithms for Intelligent Systems, https://doi.org/10.1007/978-981-13-7564-4_35

In order to develop the parallel corpus research community has been trying to develop tools such as OmegaT[1] and zanata[2] to facilitate translators in their translation task. This translation task is highly memory intensive and time-consuming. Tools like OmegaT are not many and also poses some challenges. For example, OmegaT is an offline system, which lacks a collaborative translation mechanism, whereas zanata is an online system, however, it does not support dictionary like Bangla. In order to overcome these limitations, in this study, we develop a tool (i.e., named as *AmaderCAT*) for the translators, that is collaborative and highly configurable for the translation task. It has the mechanism for crowd translation. In this paper, we report the implementation details of this tool and present some preliminary results.

Our main contributions include (1) parallel corpus development tool *AmaderCAT*, which is open source and made publicly available,[3] (2) the developed parallel corpus, which is released for research purpose.[4]

The rest of the paper is organized as follows. In Sect. 2, we present related work that is relevant to this task in terms of tools and Bangla Machine translation systems. Next, we discuss the implementation details of the tool in Sect. 3. In Sect. 4, we present the details of the dataset that we developed. We discuss the findings in Sect. 5. Finally, we conclude the paper in Sect. 6.

2 Related Work

In this Section, we first discuss the related work that has been done for developing parallel corpus. Then, we discuss related work for Bangla Machine Translation.

In Table 1, we provide a selected list of tools that are commonly used for parallel corpus development. We listed them based on the license, online/offline, and collaborative functionalities. Listing these tools helped us in understanding why it is important to develop a new tool, which can provide additional benefits to the research community.

Typically, most of the translation systems use Translation Memory (TM)[5] [22], which facilitates the translation tasks by automatically suggesting the matched translation. The matching is typically done by fuzzy matching.[6] Translation memory systems actively support the translation process by automatically suggesting existing translations and terminology [12, 22]. TM helps the translation process faster by

[1] http://omegat.org.

[2] http://zanata.org.

[3] https://github.com/AridHasan/Data-Collection-System-for-Machine-Translation.

[4] https://github.com/AridHasan/Data-Collection-System-for-Machine-Translation/tree/master/data.

[5] TM is a database consisting of source and target language pairs. It is usually stored in the database while translating the text corpus by the translators.

[6] Fuzzy matching is an approximate matching approach that tries to find a segment of matched translation by matching them with previously translated sentences. The segment can be a phrase or the whole sentence.

Table 1 Most widely used parallel corpus development tools

Name	License	Online/Offline	Collaborative
SDL Trados Studio [10, 20, 21, 25]	Commercial	Online/Offline	Yes
OmegaT	GPLv3+	Offline	No
Zanata	GNU	Online	Yes
Sketch Engine	Commercial/Free	Online	No

showing the meaning of a matching sequence in different sentences and the meaning may be different from each other. It also increases the number of the translated sentences in Computer-Assisted Translation System by assisting and accelerating the translation process [17].

The tools mentioned in Table 1 consists of different licenses such as commercial, free, and open source. SDL Trados Studio is a commercial software while Sketch Engine provides both free and commercial support. Since OmegaT and Zanata are open source, therefore, we have been interested to use and compare our system with these two tools. OmegaT has been one of the widely used tools [11], which also utilizes a Translation Memory (TM) while translating the sentences in order to facilitate the translators. At the same time, it uses a terminology, which also helps the user while translating. The limitation of OmegaT is that it is an offline system, therefore, it is difficult to manage translations across translators. It also poses challenges while distributing the translation task for a large number of crowd translators. Zanata is another open-source and collaborative tool, however, the limitation is that it lacks dictionary support for the language other than English.

In comparison with them, our proposed system uses verified translated sentences as Translation Memory. In addition, we use suffix striping using a predefined set of rules. The TM checks all the possible matches with fuzzy matching at run-time and shows all the matching source words with their meaning to the translators. The suffix stripping checks the longest possible suffix and remove the unnecessary suffix from the root words. If longest possible suffix exists then it attach some vowels at the end of the words for getting actual root words to find the meaning for the root word in the glossary. After that it shows the meaning to the translators. We evaluated our system by developing a Bangla–English parallel corpus. However, this will work for any language pair.

The works related to the Bangla Machine translation system are relatively few compared to the other languages. In [4], authors used Context-Free Grammars (CFGs) to develop English to Bangla Machine Translation system, which uses sentence construction rules for translating English to Bangla. The authors in [3] used transfer-based machine translation approach for Bangla to English machine translation system [3], which can translate simple assertive sentences. Ahmed et al. defined a set of mapping rules for the development of an MT system, which utilizes Bangla grammatical structure from an input English sentence and can translate affirmative sentences [1].

3 System Architecture

3.1 Collaborative Platform Development

In Fig. 1, we present the system architecture of our parallel corpus development collaborative platform, named as *AmaderCAT*. As a starting point of the process admin uploads source sentences and glossary files, which are stored in the database. Admin has the role of creating credentials for the translators and any modification of the translator's roles are stored in the database.

On the translation interface users (i.e., translators) needs to login to the system. While translators start translating the TM database will grow and starts showing the suggestions using fuzzy matching. In our system, we implemented edit-distance for fuzzy matching [9]. The mismatch between the source segment and the TM segment is easy to detect [16] using edit distance. In the edit distance method, we considered a distance of zero to match each word.

Glossary suggestion is one of the most important tools for Computer-Assisted Translation. Our implementations support uploading a glossary for any language, which is typically done by a project admin. Therefore, if the project admin uploads a glossary and it is present in the system then the system will use it for further suggestions. We used fuzzy approximate string matching for the glossary suggestions. These automatic suggestions will help translators to translate the sentences, which our system stores in the database for the evaluation and verification. The verification process is included in our system so that expert translators can verify the translations of the crowd (i.e., nonexpert) translators. Upon verification of the translated sentences, they will be stored in the database and TM.

Additional care needed to be taken into consideration for morphologically rich languages such as Bangla. There are a lot of inflected forms for a single word. It is

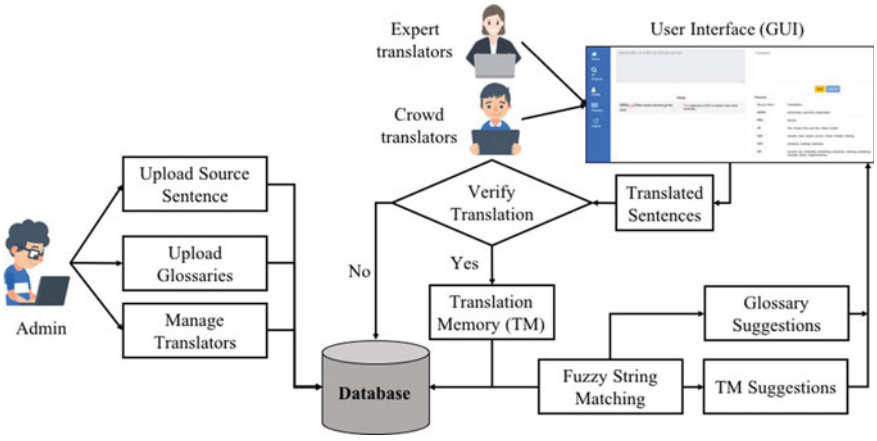

Fig. 1 System architecture of our parallel corpus development platform

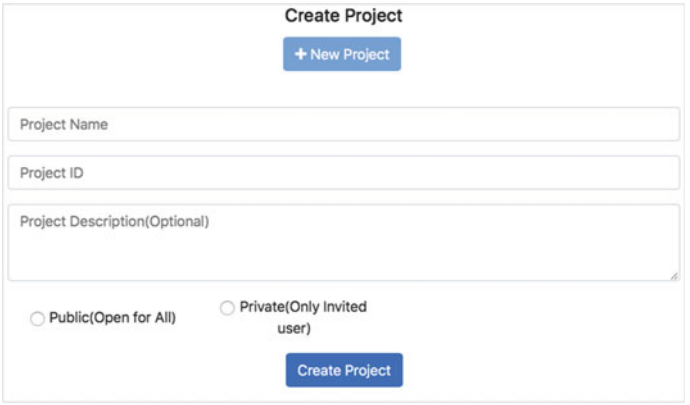

Fig. 2 User interface of the project creation process

quite difficult to determine the stem of the word from the inflected words due to the inflectional and derivational variant forms of a word [8, 18]. For clarity, we present a few examples in Example 1, 2 and 3. As discussed in [23], in many cases we need to add a letter া the end of the extracted root form that is achieved after chopping off the suffix. According to Rabia [23], this type of transformation occurs due to the causative application in Bangla verbs.

Example 1 করেছিলেন - েছিলেন -> কর + া = করা

Example 2 বলেছিল - েছিল -> বল + া = বলা

Example 3 খেলেছিলাম - েছিলাম -> খেল + া = খেলা

Root words are extracted for getting the meaning of the inflected words and showing the meaning to the translators. For glossary and TM suggestion, we listed more than 200 suffix words. Noun, verb, and person suffix were included in the list. We used a rule-based Bangla stemmer for getting root words from inflected words [18]. These root words are used for finding glossary suggestions from glossary items and TM suggestions from TM using fuzzy matching [9].

In Fig. 2, we present the user interface of the project creation page, which requires a project name, a unique project identity, a project description text box and an option for selecting the type of project. The description text box enables users to describe the purposes and provide a brief description of the project. There are two project types, such as public and private. Public project type enables all the translators who are registered in the system to contribute to the project. Whereas with the private project type only invited translators can contribute, which is managed by the project creator or project admin. The invited translators can register by accepting the email invitation or can register themselves. If the invited person is not registered in the system then the user will be asked to register. The translators are two types such as beginner and expert. The user types are not the same for every project. Beginner

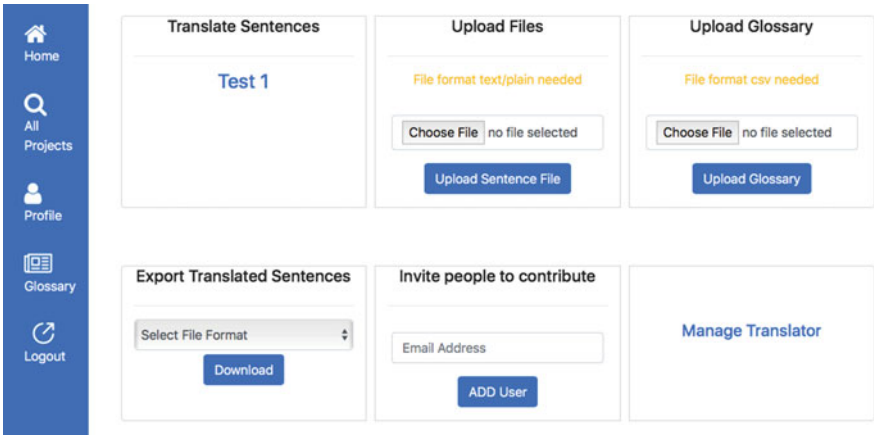

Fig. 3 Project settings for the project creator/admin

translators can only translate sentences and the expert translators can translate and verify translated sentences. In terms of translation, the translators can translate both public and invited projects, however, in order to contribute to a private project the admin of the private project needs to approve who can contribute to the project.

In Fig. 3, we present another user interface, which demonstrates how a project needs to be set up. It requires a file containing source sentences, which needs to be translated and the file format must be in plain text. Project creators can upload glossary files for the project but the glossary files are not mandatory and the file format must be in csv (i.e., comma separated) format. If no glossary file is present in the project then the system will use system glossary to show glossary suggestion. For the system, we developed a unified glossary by collecting data from different dictionaries, which we use as a system glossary. Currently, we only implemented Bangla–English glossary list as a system glossary. We plan to include more language pairs in our future work.

In Fig. 4, we present another user interface of our system to demonstrate how the translation process works. While opening the translation interface the system shows a sentence to be translated. At the same time, it shows glossary and TM suggestions. Such suggestions can facilitate the translation works. In the glossary section of this Figure, we also observe that for some words there are inaccurate translations. Those suggestions appear due to the limitations of our rule-based stemmer. We plan to improve its performance in future.

Once a translation process is completed by a beginner it can be verified by an expert and can be approved, which could be the final translated sentences. At the same time, the verified translated sentences will be saved in the TM database. In order to use the translated corpus for a further task such as Machine translation, the project creator can download the translated sentences whenever needed.

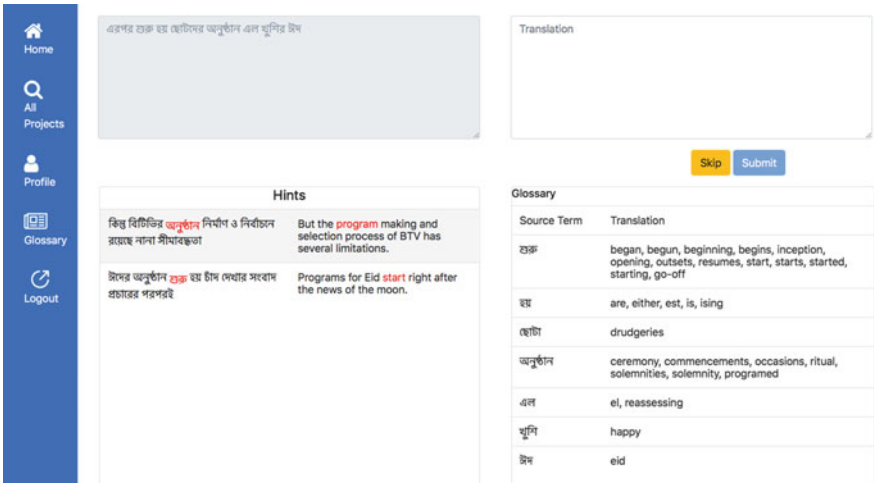

Fig. 4 Translation editing interface for the translator

3.2 System Evaluation

In order to understand the usability of our system we evaluated the system using *cognitive walkthrough approach* [19, 24]. It is an approach in which one or more evaluators perform a series of tasks to evaluate the system in terms of understanding the system's learnability for new users. For the experiment, we invited few students from the English department to participate in the evaluation and seven students have participated. By following the *cognitive walkthrough approach*, we designed the a set of questions listed below. Five tasks were given among the students to perform the tasks by using our system and asked them to answer the questions. We designed the task using the Likert scale method [2]. The scoring of the Likert scale was calculated between -2 and 2.

1. The application has a user-friendly interface?
2. The application is easy to navigate?
3. The application allows the user to upload files easily?
4. You tried and achieve the right outcome?
5. Correct action available to you to reach your goal?
6. The outcome you expect to achieve comes from the right action?
7. Correct action is performed and the progress is being made towards your intended outcome?

While performing the task we calculated the task completion time. At the same time we compute *task completion rate* and *number of error per task* using the Eqs. (1) and (2). To understand the effectiveness of our system, we run the same set of experiments using *OmegaT*.

Table 2 Evaluation of the system. Task completion rate higher is better, error rate lower is better and average usability score higher is better

Metric	AmaderCAT	OmegaT
Task completion time	4.8 min	4.95 min
Task completion rate	94.29	85.71
Error rate	0.23	0.34
Average usability score	64	53

$$Completion\ Rate = \frac{Number\ of\ Task\ Completed}{Total\ Task} \times 100\% \tag{1}$$

$$Number\ of\ Errors\ Per\ Task = \frac{Number\ of\ Error}{Total\ Task} \tag{2}$$

In Table 2, we present a comparison results for the two systems. From the table, it is clearly visible that *AmaderCAT* gets higher usability score than OmegaT. In general out of seven students six students recommended *AmaderCAT* over *OmegaT*. It is needed to mention that the number of users for this evaluation experiment are low, therefore, we plan to run another experiment with more users to prove that our results are statistically significant.

4 Data Collection

The main motivation in developing *AmaderCAT* system was to develop parallel corpora for the Machine Translation. We believe that using our preliminary version of the system one can develop a parallel corpus for any language pair such as Bengali to English, English to Bengali, English to Spanish, etc. To translate the corpus using our system we randomly selected 2,000 sentences from Bangla newspapers. For translating those sentences, we assigned few sentences to the students of English department of Daffodil International University to translate with pen and paper. For translating the rest of the sentences, we invited a few students to translate those sentences using our system. As a result, we got 1,800 translated sentences out of 2,000 Bengali sentences.

In our parallel corpus, we have ∼15,064 Bengali words in 1800 sentences and ∼19,102 English words in 1,800 sentences where the maximum numbers of words for both Bengali and English in a sentence are 12 and 30, respectively. The average number of words in a Bengali sentence is 7.53 words per sentence where an average number of words English sentence is 10.61 words per sentence. In our corpus, we have 6,640 unique Bengali words and 4931 unique English words. We ignore the inflection forms while calculating unique words for Bengali dataset.

5 Discussion

The parallel corpus development is a challenging task due to the fact that it is memory intensive, requires a high cognitive load. In order to make the task easier for the translators, it is necessary to provide them with useful tools that can facilitate them in the translation process. Even though there have been publicly available tools, however, there are some limitations in order to use them in a collaborative manner with the crowd translators. As a result, we developed *AmaderCAT*, which will significantly help the research community. As this is a very early version of our work, hence, we hope to improve it in the future. In addition, we also make the translated parallel corpus available for the research community.

6 Conclusion

In this study, we present a parallel corpus development tool –*AmaderCAT*, which is open source and publicly available. The main functionality of the system is that it is collaborative and can be used for crowd translation work. It has a great potential for a large-scale parallel corpus development work. In order to check how the system works in terms of usability, we run experiments using a *cognitive walkthrough approach*, which proves that *AmaderCAT* has more potential compared to *OmegaT*. We have completed our first set of data collection, which we made public for Bangla Machine translation research. We believe more functionalities can be added to the system to make it more useful, which we plan to do in the future. Our forthcoming work also includes more data collection and developing a Bangla Machine translation system.

Acknowledgements We would like to extend our sincere thanks to A S M Humayun Morshed from Daffodil International University and students from Department of English of the same University for helping us with the data collection task. We also would like to thank our anonymous reviewers for their detailed and constructive comments.

References

1. Ahmed S, Rahman MO, Pir SR, Mottalib M, Islam MS (2003) A new approach towards the development of English to Bangla machine translation system. In: International conference on computer information and technology (ICCIT). pp 360–364
2. Allen IE, Seaman CA (2007) Likert scales and data analyses. Qual Prog 40(7):64–65
3. Asaduzzaman S, Ali MM (2003) Transfer machine translation-an experience with Bangla English machine translation system. In: Proceedings of the international conference on computer and information technology (ICCIT). Bangladesh
4. Ashrafi SS, Kabir MH, Anwar MM, Noman A (2013) English to Bangla machine translation system using context-free grammars. Int J Comput Sci Issues (IJCSI) 10(3):144
5. Brown PF, Pietra VJD, Pietra SAD, Mercer RL (1993) The mathematics of statistical machine translation: parameter estimation. Comput Linguist 19(2):263–311

6. Chiang D (2005) A hierarchical phrase-based model for statistical machine translation. In: Proceedings of the 43rd annual meeting on association for computational linguistics. Association for Computational Linguistics, pp 263–270
7. Cho K, Van Merriënboer B, Bahdanau D, Bengio Y (2014) On the properties of neural machine translation: encoder-decoder approaches. arXiv:1409.1259
8. Das S, Mitra P (2011) A rule-based approach of stemming for inflectional and derivational words in Bengali. In: 2011 IEEE Students' technology symposium (TechSym). IEEE, pp 134–136
9. Dobrišek S, Žibert J, Pavešić N, Mihelič F (2008) An edit-distance model for the approximate matching of timed strings. IEEE Trans Pattern Anal Mach Intell 4:736–741
10. Escartín CP (2012) Design and compilation of a specialized Spanish-German parallel corpus. In: LREC. pp 2199–2206
11. Harshawardhan R, Augustine MS, Soman K (2011) Phrase based English-Tamil translation system by concept labeling using translation memory. Int J Comput Appl 20(3):1–6
12. Hummel J, Knyphausen I (2006) Method and apparatus for processing source information based on source placeable elements. US Patent 7,020,601, 28 Mar 2006
13. Koehn P (2005) Europarl: a parallel corpus for statistical machine translation. MT summit 5:79–86
14. Koehn P (2009) Statistical machine translation. Cambridge University Press, Cambridge
15. Koehn P, Hoang H, Birch A., Callison-Burch C, Federico M, Bertoldi N, Cowan B, Shen W, Moran C, Zens R, et al (2007) 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
16. Koehn P, Senellart J (2010) Convergence of translation memory and statistical machine translation. In: Proceedings of AMTA workshop on MT research and the translation industry. pp 21–31
17. Lagoudaki E (2006) Translation memories survey 2006: users perceptions around tm use. In: proceedings of the ASLIB international conference translating and the computer, vol 28. pp 1–29
18. Mahmud MR, Afrin M, Razzaque MA, Miller E, Iwashige J (2014) A rule based Bengali stemmer. In: 2014 International conference on advances in computing, communications and informatics (ICACCI). IEEE, pp 2750–2756
19. Nielsen J (1994) Usability inspection methods. In: Conference companion on human factors in computing systems. ACM, pp 413–414
20. Ruiz Yepes G, et al (2011) Parallel corpora in translator education
21. Skadiņš R, Puriņš M, Skadiņa I, Vasiļjevs A (2011) Evaluation of SMT in localization to under-resourced inflected. In: 15th international conference of the European association for machine translation. pp 35–40
22. Somers H (2003) Translation memory systems. Benjamins Transl Libr 35:31–48
23. Ummi RS (2013) A rule-based stemmer for Bangla verbs. PhD thesis, Independent University
24. Wharton C (1994) The cognitive walkthrough method: a practitioner's guide. Usability inspection methods
25. Zampieri M, Vela M (2014) Quantifying the influence of MT output in the translators' performance: a case study in technical translation. In: Proceedings of the EACL 2014 workshop on humans and computer-assisted translation. pp 93–98