ISSN No. 0976-5697

Volume 8, No. 5, May-June 2017

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

Pre-processing Steps on Bilingual Corpus for SMT

Abhijit Paul
Department of Computer Science
Assam University
Silchar, India

Prof. Bipul Syam Purkayastha
Department of Computer Science
Assam University
Silchar, India

Abstract: Correct pre-processing techniques on bilingual parallel corpus (text translated into two different languages) play an important role in the Statistical Machine Translation (SMT) systems. The purpose of this paper is to examine the effectiveness of the pre-processing techniques that are used for training the translation systems. This paper also discusses the various pre-processing steps used for SMT. Experimental results of the pre-processing steps on Technology Development for Indian Languages (TDIL) '**English-Nepali'** parallel corpus (on tourism domain) of size over 4K sentences are discussed.

Keywords: SMT, Parallel Corpus, Computational Linguistics, Domain

1. INTRODUCTION

Statistical Machine Translation (SMT) systems have gained much interest in recent years. The advantage of this system is that it is fully automatic and requires significantly less human effort than traditional rule based approach. However, it requires sentence-aligned parallel corpus for each language pair and cannot be used for language pairs for which such corpora do not exist [1]. SMT translates based on the information what it has trained from parallel corpus. This corpus consists of two texts, each of which is the translation of the other. Parallel corpus is also useful for other natural language processing tasks such as information retrieval, word sense disambiguation, etc. Training the translation model component in SMT requires large parallel corpora for the parameters to be estimated [2].

Apart from large parallel corpus, pre-processing on the training data can help to build & improve the SMT model components (language model, translation model and decoder) quality and by extension the performance of an SMT system. Pre-processing is basically to process something before it is being processed by something else. In computer science a pre-processor is nothing but a program or a system that processes its input data to produce output that is used as input to another program. In this paper important pre-processing steps for SMT like sentence splitting, tokenization, truecasing and cleaning techniques have been discussed. Other pre-processing like morphological analysis, part of speech tagging, word sense disambiguation, name entity recognition, etc are there for traditional MT, but in this pre-processing technique only SMT basic pre-processing steps are taken into consideration.

TDIL English-Nepali parallel corpus [3] has been used for testing the system, which is available for the purposes of MT research.

2. RELATED WORK

In this section a few number of journal papers have been discussed to get the idea of suitable pre-processing techniques for MT system.

In the paper [4], tokenization, truecasing and cleaning pre-processing techniques are used. The main aim of the paper was development of English to Malayalam Machine Translation using Phrase Based Statistical Approach. For the translation work, training data is provided in a sentence aligned (one sentence per line) format, in two files, one for the English sentences and one for the Malayalam sentences. English sentences are stored in a file with **.en** extension and the corresponding Malayalam translations are stored in a file with **.ml** extension with one sentence per line. As Malayalam is written in non-Latin script, the Malayalam corpus is Romanized first before pre-processing.

In the paper [5], a pre-processing method has been discussed that reorders source words according to the corresponding target word order suggested by an initial word alignment for statistical machine translation. This paper mentioned a point that the translation problem can be reduced through corpus pre-processing steps that perform grouping and splitting of words.

In the book one chapter Alignment by Agreement [6] for pre-processing steps, authors have lowercased all words, then they used the validation set and the first 100 sentences of the test set as their development set to tune their models. This chapter mainly focuses on word alignment as important pre-processing steps for MT.

In the paper [7], pre-processing steps for SMT components like Language Model and Translation Model include i) Normalization, which converts all words of source and target language to upper or lower cased in all sentences, ii) Tokenization, for all data in the both language corpus data by inserting spaces between words and punctuation, iii) True-casing, by generating probabilities for all words in the parallel corpus and building a model. Truecasing is applied on both languages (target as well as source). Another pre-processing step is Cleaning, data by delete long sentences, which are longer than specific number or remove empty sentence or misaligned sentences, which can affect the translation quality.

The paper [9] processed the corpus through appropriate filters for normalization. Stanford tokenizer has been used for tokenizing English Corpus. For normalization, truecase.perl model provided in MOSES toolkit has been used. Hindi Corpus Normalization has been done using NLP Indic Librar. This paper also performed Data Split.

To prepare the data for training, the author [8] has been performed some pre-processing steps. The pre-processing was as follows:
1) Tokenize the Assamese and English corpus.
2) Lowercase of the English corpus.
3) Cleaning the data, i.e. removing extra spaces, empty lines and lines that are too short or too long.

3. COMMON PRE-PROCESSING STEPS FOR MACHINE TRANSLATION

Pre-processing is the process of something before it is being processed by something else. In the field of NLP, a pre-processor is a program that processes its input data to produce output that is being used as input to another program. Following are the widely used pre-processing steps for machine translation.

i. Tokenization: Tokenization is the processes of getting the most constitute part, i.e. tokens of the sentence. It is considered as the first pre-processing steps for any NLP task.

ii. Stemming: Stemming is also one of the prerequisite steps for Text Mining, Spelling Checker as well as Machine Translation. Stemming is the process of finding the root word of a word along with its other affixes viz. infix and suffix [**10**].

iii. Morphological Analysis: For Machine Translation between any language pair, the amount of parallel data and the language differences, mainly the morphological richness and word order differences can affect the performance of the Machine Translation **[11]**. Morphological analysis is the process of analyzing the different morphemes along with all the grammatical features of the word in a sentence.

iv. Part of Speech Tagging: Part of speech tagging is the process of identifying the lexical class marker or part of speech category of a sentence. Training part of speech tagger is useful to build machine translation system for less resourced language pairs **[12]**.

v. Parsing: Parsing, transforms one natural language sentence to parse tree or syntax tree. The elements in the parse tree are phrases (build using part of speech category) and tokens. Parsing is the useful technique for Machine Translation to identify the source language structure i.e. for pre-ordering as well as target language structure i.e. for post -ordering.

vi. Name Entity Recognition: Named entities create serious problems for machine translation (MT) systems and often cause translation failure. Name Entity Recognition is the process of identification of named entities (NEs) in the source language sentence **[14]**.

vii. Word Sense Disambiguation: Word Sense Disambiguation is the process of determining the sense of a word from contextual features. The correct sense of a word can improve the performance of the machine translation [15].

4. PROPOSED AUTOMATED PRE-PROCESS STEPS ON PARALLEL DATA

As the main purpose of doing this work is to develop a machine translation system using statistical approach so in this paper important SMT pre-processing steps are implemented. Other pre-processing steps like identifying the sense of the word, arrange the structure of the word order will be done by the core components (LM, TM and DECODER) of the SMT system. The output of this pre-processing work is the English and Nepali pre-processed file, which will be useful for developing English-Nepali SMT system. This section gives an overview of the proposed automated pre-processing steps. Figure 1 below shows the pre-processing steps.

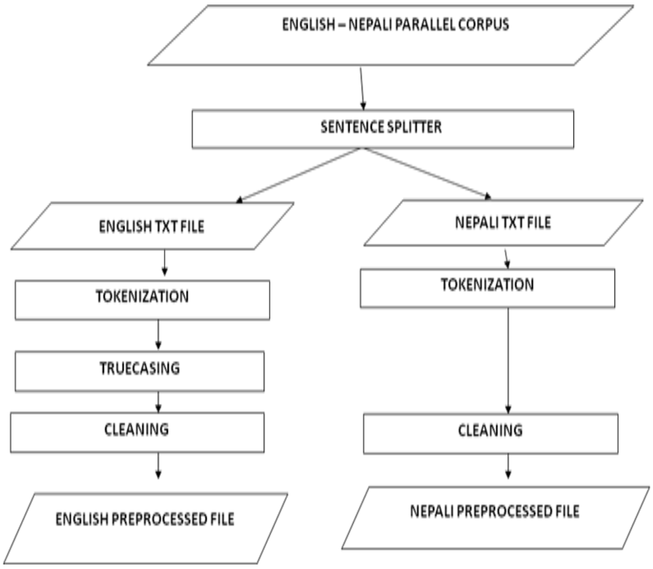


Fig 1: Pre-processing steps on bilingual corpus

Step1. Sentence Splitter: - The corpora that which are collected from TDIL consist of 51 files in .XL format and total number of sentences in these files are 8830 (4415 English sentences and 4415 Nepali sentences). The sentence splitter is a small program written in JAVA which takes these 51 files as an input and produce two separate text files, one file consists of English sentences & other consist of translated Nepali sentences.

Step2: Tokenization: - One of the basic text processing steps is tokenization, the breaking up of raw text into words. This means that spaces have to be inserted between words and punctuation. Though English language Tokenizer is easily available in the web but for Nepali language or other Indian languages, it is not. Also English Tokenizer does not work properly for Indo-Aryan languages (e.g.: Nepali) due to its multiple characteristics and compounds words. So a code is written using Perl scripts for tokenization of Nepali sentence. And for English sentences, Moses inbuilt Tokenizer scripts have been used.

Steps3. Truecasing: - Related to tokenization, truecasing is the issue of words that occur in lowercased or uppercased form in the text. The items house, House and HOUSE may occur in a large text collection, in the middle of a sentence, at the beginning of a sentence, or in the headline, respectively. It is the same word, so it is useful to normalize the case, i.e. convert to their most probable casing, typically by lowercase or truecasing. This helps the system to reduce data sparsity. The letter preserves uppercase in names, allowing the distinction between Mr. Fisher and a fisher. Since there is no concept of uppercase and lowercase in the

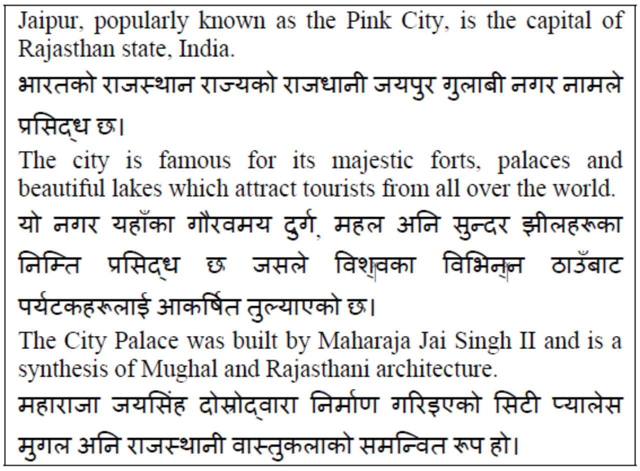
Nepali language so applying truecasing on Nepali language is meaningless. The output of English Tokenizer is the input file of English truecasing.

Step4. Cleaning:- Cleaning is the process of removing the long sentences, empty sentences and misaligned sentences from truecasing files. Long sentences, empty sentences and misaligned sentences can cause problems with the training pipeline. Along with English truecasing files, Nepali Tokenizer files need to clean for smooth training.

These pre-processed English and Nepali files will help to develop the SMT basic components i.e. language model, target model and decoder.

5. EXPERIMENTAL RESULTS

This section shows and discusses the experimental result. The format of the parallel corpus is (only four sentence pair is shown):-



Jaipur, popularly known as the Pink City, is the capital of Rajasthan state, India.

भारतको राजस्थान राज्यको राजधानी जयपुर गुलाबी नगर नामले प्रसिद्ध छ।

The city is famous for its majestic forts, palaces and beautiful lakes which attract tourists from all over the world.

यो नगर यहाँका गौरवमय दुर्ग, महल अनि सुन्दर झीलहरूका निम्ति प्रसिद्ध छ जसले विश्वका विभिन्न ठाउँबाट पर्यटकहरूलाई आकर्षित तुल्याएको छ।

The City Palace was built by Maharaja Jai Singh II and is a synthesis of Mughal and Rajasthani architecture.

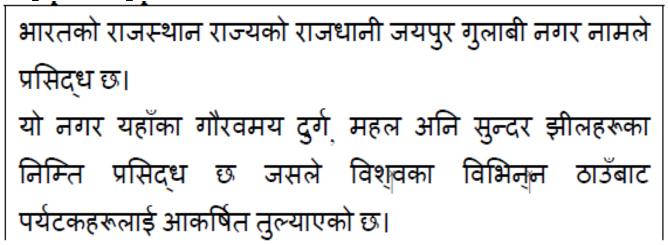
महाराजा जयसिंह दोस्रोद्वारा निर्माण गरिइएको सिटी प्यालेस मुगल अनि राजस्थानी वास्तुकलाको समन्वित रूप हो।

i. Results of sentence splitter

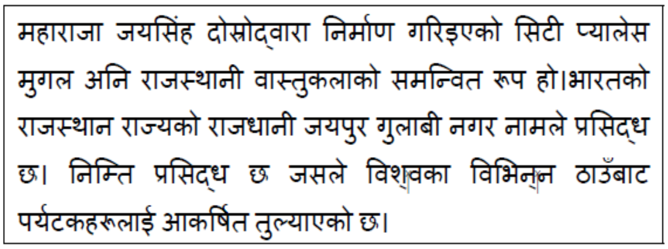
English Corpus:-

Jaipur, popularly known as the Pink City, is the capital of Rajasthan state, India.
The city is famous for its majestic forts, palaces and beautiful lakes which attract tourists from all over the world.
The City Palace was built by Maharaja Jai Singh II and is a synthesis of Mughal and Rajasthani architecture.

Nepali Corpus:-



भारतको राजस्थान राज्यको राजधानी जयपुर गुलाबी नगर नामले प्रसिद्ध छ।
यो नगर यहाँका गौरवमय दुर्ग, महल अनि सुन्दर झीलहरूका निम्ति प्रसिद्ध छ जसले विश्वका विभिन्न ठाउँबाट पर्यटकहरूलाई आकर्षित तुल्याएको छ।



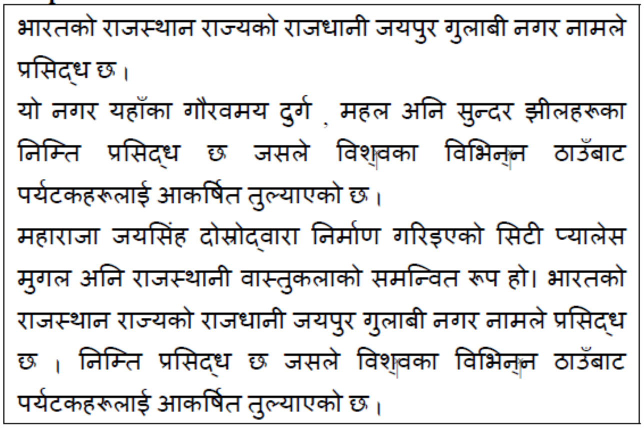
महाराजा जयसिंह दोस्रोद्वारा निर्माण गरिइएको सिटी प्यालेस मुगल अनि राजस्थानी वास्तुकलाको समन्वित रूप हो।भारतको राजस्थान राज्यको राजधानी जयपुर गुलाबी नगर नामले प्रसिद्ध छ। निम्ति प्रसिद्ध छ जसले विश्वका विभिन्न ठाउँबाट पर्यटकहरूलाई आकर्षित तुल्याएको छ।

ii. Results of tokenization:-

English tokenized File:-

Jaipur, popularly known as the Pink City , is the capital of Rajasthan state , India .
The city is famous for its majestic forts, palaces and beautiful lakes which attract tourists from all over the world.
The City Palace was built by Maharaja Jai Singh II and is a synthesis of Mughal and Rajasthani architecture.

Nepali tokenized File:-



भारतको राजस्थान राज्यको राजधानी जयपुर गुलाबी नगर नामले प्रसिद्ध छ ।
यो नगर यहाँका गौरवमय दुर्ग , महल अनि सुन्दर झीलहरूका निम्ति प्रसिद्ध छ जसले विश्वका विभिन्न ठाउँबाट पर्यटकहरूलाई आकर्षित तुल्याएको छ ।
महाराजा जयसिंह दोस्रोद्वारा निर्माण गरिइएको सिटी प्यालेस मुगल अनि राजस्थानी वास्तुकलाको समन्वित रूप हो। भारतको राजस्थान राज्यको राजधानी जयपुर गुलाबी नगर नामले प्रसिद्ध छ । निम्ति प्रसिद्ध छ जसले विश्वका विभिन्न ठाउँबाट पर्यटकहरूलाई आकर्षित तुल्याएको छ ।

The table 1 below shows the findings of the tokenizer process in terms of amount of words encountered in the corpus.

Table 1: Results of tokenization

	Amount of words
Normal English Corpus	93574
Tokenized English Corpus	94223
Normal Nepali Corpus	76414
Tokenized Nepali Corpus	76617

iii. Results of truecasing:-

English truecased File:-

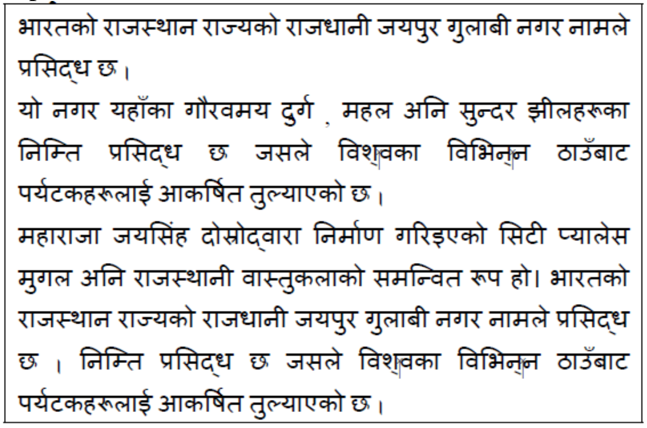
jaipur , popularly known as the pink city , is the capital of rajasthan state , india .
the city is famous for its majestic forts , palaces and beautiful lakes which attract tourists from all over the world .
the city palace was built by maharaja jai singh II and is a synthesis of mughal and rajasthani architecture .

iv. Results of cleaning:- Cleaning process perform only to cut down the length of the sentence and remove the unusual space between words and between sentence also.

English cleaned File:-

jaipur , popularly known as the pink city , is the capital of rajasthan state , india .
the city is famous for its majestic forts , palaces and beautiful lakes which attract tourists from all over the world .
the city palace was built by maharaja jai singh II and is a synthesis of mughal and rajasthani architecture .

Nepali cleaned File:-



भारतको राजस्थान राज्यको राजधानी जयपुर गुलाबी नगर नामले प्रसिद्ध छ ।
यो नगर यहाँका गौरवमय दुर्ग , महल अनि सुन्दर झीलहरूका निम्ति प्रसिद्ध छ जसले विश्वका विभिन्न ठाउँबाट पर्यटकहरूलाई आकर्षित तुल्याएको छ ।
महाराजा जयसिंह दोस्रोद्वारा निर्माण गरिइएको सिटी प्यालेस मुगल अनि राजस्थानी वास्तुकलाको समन्वित रूप हो। भारतको राजस्थान राज्यको राजधानी जयपुर गुलाबी नगर नामले प्रसिद्ध छ । निम्ति प्रसिद्ध छ जसले विश्वका विभिन्न ठाउँबाट पर्यटकहरूलाई आकर्षित तुल्याएको छ ।

Results of Language Model (LM):

A language model gives the probability of the sentence. The probability is computed using n-gram model. Language model can be considered as the computation of the probability of single word given all the preceded it in a sentence. Language model is trained using the final preprocessed (Nepali cleaned) file as well as without preprocessed Nepali file. Both these two models are tested using 5 input sentences and result is shown below. The model gives the correct probability if the file is preprocessed well.

Table 2: Results of LM

	Accuracy of LM (%)
Using Pre-processing	100%
Without Using Pre-processing	73%

Results of Translation Model (TM):

The translation model computes the conditional probability. It also constructs the phrase table, which is useful for translation. Phrase table contains phrases or segments of English words along with their meaning in Nepali. Two translation models are prepared, one using preprocessed English-Nepali files containing 60 and 66 tokens respectively and other using normal English-Nepali files containing 50 and 54 respectively. The models are checked manually and results are shown below.

Table 3: Results of TM

	Correct Meaning Transferred
Using Pre-processing	94%
Without Using Pre-processing	66%

6. CONCLUSION & FUTURE DIRECTION

Though the machine translation is the earliest applications of NLP and so many MT systems are available in India for translating text from Indian to English, English to Indian and Indian to Indian languages, but the goal of achieving error-free translation that reads fluently in target languages is still far off; limited success has been achieved within a restricted domain. Pre-processing is one of the important steps for developing a correct machine translation system. In this paper, different pre-processing techniques along with its experimental results have been discussed. And it finds that the results of basic components of SMT (LM, TM) are improved using the pre-processing techniques.

After performing pre-processing work on parallel corpus one can easily design and develop the basic components of SMT. In case of correct tuning (an important step of SMT), correct pre-processing is required. These things can be considered as the future direction of this research work.

7. ACKNOWLEDGMENT

Thanks to entire research colleague, Department of Computer Science, Assam University Silchar, India for their support and inspiration throughout the duration of the work. Finally thanks go to god and all our well wisher.

8. REFERENCE

[1] Siddiqui T., Tiwary U.S.. Natural Language Processing and Information Retrieval, Oxford Publication, 2008.

[2] Taghipour, Kaveh, Afhami N., Khadivi S. and Shiry S. (2010) "A discriminative approach to filter out noisy sentence pairs from bilingual corpora", Telecommunications (IST), 5th International Symposium on 2010 : 537-541.

[3] www.tdil.mit.gov.in

[4] Nithya B., Shibily J.. A Hybrid Approach to English to Malayalam Machine Translation, International Journal of Computer Applications (0975 – 8887) Volume 81 – No.8, November 2013.

[5] Holmqvist M, Stymne, Ahrenberg L., Merkel M., Alignment-based reordering for SMT firstname.lastname@liu.se

[6] Book: - Liang P., Taskar B., Klein S., Alignment by Agreement.

[7] Ahmed G. M. ElSayed , Ahmed S. Salama and Alaa El-Din M. El-Ghazali, A Hybrid Model for Enhancing Lexical Statistical Machine Translation (SMT) , IJCSI International Journal of Computer Science Issues, Volume 12, Issue 2, March 2015 , ISSN (Print): 1694-0814 | ISSN (Online): 1694-0784 , www.IJCSI.org

[8] Das P., Baruah K.K. Assamese to English Statistical Machine Translation Integrated with a Transliteration Module , International Journal of Computer Applications (0975 – 8887) Volume 100– No.5, August 2014

[9] Dungarwal P., Chatterjee R., Mishra A., Kunchukuttan A., Shah R., Bhattacharyya P., *The IIT Bombay Hindi, English Translation System at WMT 2014* . Workshop on Machine Translation **(WMT 2014)**. 2014.

[10] Paul A., Dey A., Purkayasth B.S., An Affix Removal Stemmer for Natural Language Text in Nepali International Journal of Computer Applications 91(6):1-4, April 2014.

[11] Koehn, P., Birch, A., and Steinberger, R. (2009). 462 Machine Translation Systems for Europe. In MT Summit XII

[12] Sánchez-Martınez, Felipe, et al. "Training part-of-speech taggers to build machine translation systems for less-resourced language pairs." Procesamiento del Lenguaje Natural (XXIII Congreso de la Sociedad Espanola de Procesamiento del Lenguaje Natural). Vol. 39. 2007

[13] Carreras, Xavier, and Michael Collins. "Non-projective parsing for statistical machine translation." Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1-Volume 1. Association for Computational Linguistics, 2009.

[14] Babych, Bogdan, and Hartley. A. "Improving machine translation quality with automatic named entity recognition." Proceedings of the 7th International EAMT workshop on MT and other Language Technology Tools, Improving MT through other Language Technology Tools: Resources and Tools for Building MT. Association for Computational Linguistics, 2003.

[15] Vickrey, David, et al. "Word-sense disambiguation for machine translation." Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing. Association for Computational Linguistics, 2005.