


Constructing High Quality Bilingual Corpus using Parallel Data from the Web

Sai Man Cheok^{1,2}, Lap Man Hoi^{1,2}, Su-Kit Tang^{1,2}^a and Rita Tse^{1,2}

¹*School of Applied Sciences, Macao Polytechnic Institute, Macao SAR, China*

²*Engineering Research Centre of Applied Technology on Machine Translation and Artificial Intelligence of Ministry of Education, Macao Polytechnic Institute, Macao SAR, China*

Keywords: Machine Translation, CNN Modelling, Bilingual Corpus, Parallel Data.

Abstract: Natural language machine translation system requires a high-quality bilingual corpus to support its efficient translation operation at high accuracy rate. In this paper, we propose a bilingual corpus construction method using parallel data from the Web. It acts as a stimulus to significantly speed up the construction. In our proposal, there are 4 phases. Parallel data is first pre-processed and refined into three sets of data for training the CNN model. Using the well-trained model, future parallel data can be selected, classified and added to the bilingual corpus. The training result showed that the test accuracy reached 98.46%. Furthermore, the result on precision, recall and f1-score is greater than 0.9, which outperforms RNN and LSTM models.


1 INTRODUCTION

Machine learning technology has been applied into many different areas, solving many difficult problems (Lin, 2021) (Chan, 2021) (Chan, 2021). Natural language processing (NLP) is also one of the areas that is commonly based on machine translation technology, which requires a high-quality bilingual corpus for efficient and accurate automatic translation (Tse, 2020) (Zin, 2021) (Cheong, 2018). The quality of bilingual corpus relies on the quality of datasets used when constructing. In corpus construction, data is generally sourced from paper articles, electronic documents, and the Web. As they are not standardized in an easily readable or pre-defined format, the processing of the sourcing data becomes complicated and time-consuming. The digitalization and proofreading of data on paper materials require significant post-processing workload. If data is collected manually, significant effort on editing work is needed. Even though it is collected electronically, the data may contain bias or errors. Proofreading is unavoidable. Therefore, web crawling becomes an efficient and effective method for collecting data for bilingual corpus. To ensure the data quality, crawled data are required to be processed appropriately before storing into the corpus.

In this paper, we propose a method to construct a high-quality bilingual corpus for machine translation systems using parallel data (articles in at least two different languages) from the Web. Assuming Chinese and Portuguese are the languages to be used, there are 4 phases, which are 1) data collection, 2) data pre-processing (cleaning, segmentation and alignment), 3) model training and 4) classification, in the construction. Figure 1 depicts the four phases in the construction of the bilingual corpus.

In phase 1, for data collection, a web crawler is commonly used to automatically crawl parallel data. There are a number of web crawling architectures available, which are hybrid crawler, focused crawlers and parallel crawlers (Cheok, 2021) (Sharma, 2015) (Cho, 2002) (Chakrabarti, 1999) (Pappas, 2012). They crawl webpages automatically from tree-structural websites for some particular information by following embedded hypertext links in pages, which are then stored in a repository for further querying.

In phase 2, three pre-processing steps are included, which are cleaning, segmentation and alignment. The cleaning is usually done by removing unnecessary or unexpected alphabets or text, and by matching regular expression between bilingual sentences. Regular expressions are logical formulas, used by rules for filtering. To improve the cleaning

 <https://orcid.org/0000-0001-8104-7887>

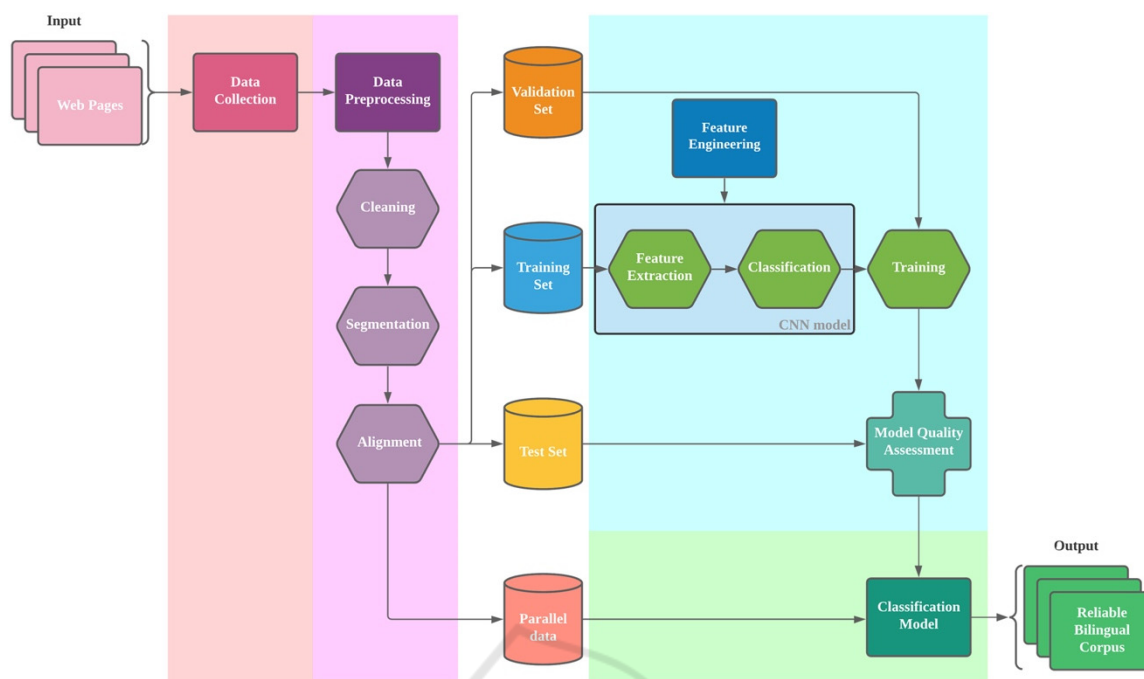


Figure 1: Four Phases in the Bilingual Corpus Construction.

efficiency and accuracy, regular expression is normally employed which removes unnecessary alphabets or text (such as tags, comments, errors, duplicates, etc.) in web pages. Moreover, to ensure the automation of data crawling effectively, bilingual alignment is needed. There are two common methods used in bilingual alignment, which are length-based and vocabulary-based (Li, 2010). The length-based method is based on simple length information. No vocabulary information is needed. Therefore, it runs fast and requires minimum storage. Vocabulary-based method is based on the vocabulary in text to achieve high accuracy rate, even though it is complex and slow.

In phase 3, feature engineering creates a segmentation model to represent the words and sentences using computer-recognized patterns in vector format for processing. Existing representation models such as Bag of Word (TF-IDF algorithm) (Zhao, 2018), and Word vectors (one-hot algorithm, word2vec algorithm, etc.) (Uchida, 2018) are commonly used. After extracting corpus features, the CNN model is selected and developed for training due to its two-dimensional structure of input data.

Finally, in phase 4, trained CNN model will be used for classification, selecting high quality parallel data to build the bilingual corpus. The training result showed that the test accuracy reaches a 98.46%. The result on precision, recall and f1-score is greater than 0.9, which outperforms RNN and LSTM models.

The remainder of the paper is structured as follows. The construction of bilingual corpus will be introduced in Chapter 2. It starts with data acquisition (Phase 1 and Phase 2), followed by the training of CNN model using three sets of data (Phase 3) in Chapter 3. Once the training is done, in Chapter 4, the Classification model will be developed, for selecting high quality parallel data to build the bilingual corpus (Phase 4). In Chapter 5, the training performance will be revealed and discussed. The CNN model over other models will be evaluated too. Finally, the remarks of this paper will be given.

2 DATA ACQUISITION

Parallel data is essential in the construction of bilingual corpus for machine translation systems. To ensure the quality of data is high, three sets of parallel data for model training are required.

2.1 Data Collection

In phase 1, collecting a large amount of parallel data is complicated and time-consuming, even it is automatic crawling from the Web. To achieve it efficiently, a web crawler for parallel data that can ensure the consistency and accuracy of the bilingual data is highly recommended (Cheok, 2021). As it is

out of scope of this paper, the crawling and processing of parallel data will not be given. In particular, as the quality of parallel data is significantly crucial to the translation quality, bilingual official websites are highly recommended.

2.2 Data Pre-processing

In phase 2, crawled parallel data are required to go through three pre-processing steps, which are Cleaning, Segmentation and Alignment. Cleaning removes unnecessary characters in the data. Segmentation divides data into individual segments, such as sentences. Alignment is the step to cross check the quality of corresponding data in other languages by comparing to the one translated from translation engine.

2.2.1 Cleaning

Crawled parallel data from websites is first refined by filtering out unnecessary elements, such as HTML tags and characters, punctuations and spaces in text, URLs and image links, etc. It happens as those elements may not appear in both lingual sentences. They even do not contribute to any meaning of the text content. For instance, the HTML characters of Chinese text apparently do not appear in Portuguese text. If sentence pairs are placed in the training set, the accuracy of the model training will be reduced, lowering the accuracy of the final corpus. Thus, in the corpus cleaning process, all unnecessary elements will be deleted, ensuring the quality of parallel data in the training set.

2.2.2 Segmentation

The Portuguese text content in an article can usually be divided into sentences by punctuation, such as period, exclamation mark, question mark and semicolon, etc. Divided sentences are then stored in pairs as a training set in the bilingual corpus. For some Latin languages, such as Portuguese, some exceptional cases are expected. One of the common examples is about the period punctuation (“.”), which does not always end a sentence. It may be an abbreviation or numbering symbol. Therefore, the punctuation period cannot be treated as an ending of a sentence in segmentation. On the other hand, in Chinese, punctuations such as period, exclamation mark, question mark and semicolon, etc., will be straightforward to serve for the word segmentation, dividing the text content into a number of sentences. It is noteworthy that, depending on the languages, special segmentation application is needed.

2.2.3 Alignment

After segmentation, data alignment is conducted which ensures the quality of parallel data, achieving a certain acceptance level. Original Chinese sentences translated into Portuguese by a third-party translation engine will be compared with the original Portuguese sentence. It is declared that translating Portuguese sentences using 3rd party translation engine for comparison instead of translating Chinese sentences is also accepted. The similarity test to be conducted (Ristad, 1998) will measure the distance value in transforming one string (the source) into another (the target) based on the minimal number of deletions, insertions, or substitutions required. If the similarity is greater than or equal to 60%, it will be stored as a high-quality bilingual corpus. This provides a buffering to balance against translation difference between translation methods (literal translation and sense-for-sense translation) in translation engines (Baker, 2001). Otherwise, it is stored in the pending corpus for subsequent processing. Figure 2 outlines the workflow of data alignment on segmented sentences.

After phase 2, three sets of parallel data will be created, which are the *training set*, validation set and test set. The training set is designed for training the Classification model so that future parallel data can be categorized accurately in the bilingual corpus. The training set goes through the feature extraction and classification in CNN model for configuration. Once the model is configured, it will be sent to training. In the training, the *validation set* is used to validate the result of the training. It ensures that the model can accurately and correctly categorize parallel data.

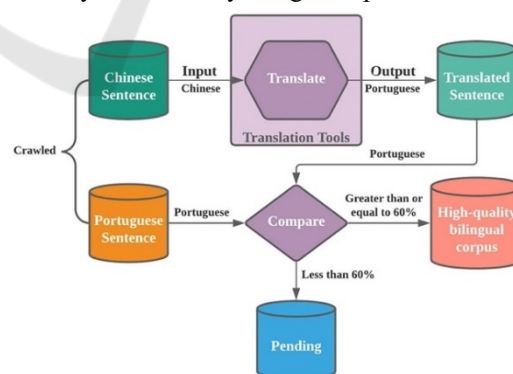


Figure 2: The Workflow of Data Alignment.

If the training result is accepted, the *test set* will be used in the Classification model for categorization. In this model, Chinese is the key in model training and the construction of the high-reliable bilingual corpus. Once the Classification model is ready, future parallel data crawled (the fourth set of data) can be categorized.

3 TRAINING FOR CLASSIFICATION

In Phase 3, one of the parallel data sets, called training set, will be further processed by extracting the features from text content. Together with another two sets of data, the CNN model is developed for training. Once the training is done with satisfactory result, the Classification model in Phase 4 will be done.

3.1 Feature Engineering

Feature engineering is a process to further manipulate the training set for improving the accuracy and efficiency of learning and recognition in Classification model. In this paper, Chinese word segmentation is employed (Zhang, 2002), which provides three particular functions, which are new word discovery, Batch segmentation and Intelligent filtering.

New word discovery. New words are excavated from the Chinese text for compilation of professional dictionaries. Editing and labelling are introduced into the word segmentation dictionary for improving the accuracy of the word segmentation system and adapting to new language changes.

Batch segmentation. Automatic recognition of new words, such as names, place names, and organization names, new word tagging, and part-of-speech tagging, can be achieved efficiently.

Intelligent filtering. Intelligently filtering and reviewing the semantics of the text content in sentences using the most complete built-in word database in China, identifying multiple variants, traditional and simplified characters, and precise semantic disambiguation can be achieved efficiently.

As the segmentation method only supports Chinese language in segmentation, other languages may need to employ other particular segmentation methods.

3.2 CNN Model

The CNN model is crucial for the accuracy of Classification result. To achieve it, Feature extraction will first be conducted, which processes the training set with the following steps.

1. Splits a sentence into multiple words;
2. Maps each word into a low-dimensional space through the word2vec embedding method;
3. Represent the text expressed by the word vector in one-dimensional;
4. Extract the maximum value of each feature vector to represent the feature after

convolution with different heights of convolution kernels.

After Feature extraction, several parameters in training engine are required to be considered for configuring the CNN model for best result. After every round of training experiment against the validation set, the CNN model will be sent to the verification process, called Model quality assessment, which tests about the model for the classification quality using the test set. If it is not accepted (result ranged below 90%), the model parameters will be revised for another round of experiment. If the result reaches above 90%, the model is done. The training result of the model can be seen in next section.

4 PERFORMANCE EVALUATION

The trained CNN model will be brought to Phase 4 as the Classification model if the training result is accepted. In the training, three types of hardware configurations have been used, which includes one server-graded computer with four GPUs, one high-end computer with one GPU and one low-end personal computer with one GPU. Table 1 summarized the configuration used in the model training.

Table 1: Hardware Configuration for the Model Training.

	NVIDIA DGX	Dell XPS	Normal PC
CPU	64-core AMD EPYC CPU	Intel Core i7-9750H	Intel Core i7-6700
GPU	4x NVIDIA A100 80 GB GPUs	NVIDIA GeForce GTX 1650 4GB	NVIDIA GeForce GTX 960 2GB
RAM	512 GB DDR4	16GB DDR4	16GB DDR4
Storage	1.92 TB NVME drive	512GB M.2 PCIe NVME SSD	256GB SSD + 2TB HDD

In software configuration, the same environment was setup, which included TensorFlow software of version 1.14.0, running in Window 10 version 2004(OS Build 19041.867), on the same data sets. For each configuration, the CNN model was trained until the result is accepted. After certain rounds of training

with parameter adjustment, a high accuracy result is obtained, as shown in Table 2.

Table 2: Summary of Training on CNN Model.

CNN model	NVIDIA DGX	Dell XPS	Normal PC
Training Loss	0.038	0.044	0.062
Training Accuracy	98.44%	96.74%	96.88%
Validation Loss	0.039	0.04	0.046
Validation Accuracy	99.20%	98.86%	99.00%
Training Time	0:03:51	0:06:36	0:08:22

As can be seen in Table 2, the result showed the CNN model works efficiently in the three machines. It takes minimum amount of time in training by NVIDIA DGX machine as it is more computing power. Besides, it is noteworthy that losses and accuracies of the CNN model for three machines are different, due to the randomness of weight initialization in neural network algorithm.

Moreover, RNN model and LSTM model are configured for comparison. Similar parameters on the same sets of data have been configured for training. In particular, the convolution kernel size and convolution kernel number in CNN model are set to be 5 and 256 respectively, and the number of hidden layers of RNN model and LSTM model is set to 2.

It is found that the training time required for NVIDIA DGX, Dell XPS and PC on RNN model was about 18 hours, 30 hours and 48 hours respectively while the training time required for LSTM model is about 42 hours, 61 hours and 98 hours respectively. Table 3 summarized the training using RNN model for each configuration and Table 4 summarized the training using LSTM model for each configuration.

Table 3: Summary of Training on RNN Model.

RNN model	NVIDIA DGX	Dell XPS	Normal PC
Training Loss	0.0033	0.0068	0.098
Training Accuracy	100.00%	100.00%	100.00%
Validation Loss	0.12	0.098	0.046
Validation Accuracy	98.13%	97.86%	97.57%
Training Time	18:14:23	30:06:48	48:36:19

Table 4: Summary of Training on LSTM Model.

LSTM model	NVIDIA DGX	Dell XPS	Normal PC
Training Loss	0.057	0.072	0.082
Training Accuracy	96.75%	96.38%	95.81%
Validation Loss	0.08	0.092	0.072
Validation Accuracy	96.70%	96.23%	97.31%
Training Time	42:28:11	61:33:15	98:29:50

The result of training on three models showed that they both achieved high training accuracy and validation accuracy at very low training and validation losses. Apparently, the amount of training time for CNN is much lower than the other models. The CNN model outperforms both RNN and LSTM models in Classification for all configurations.

With the high accuracy rate achieved in the training, the three models are further tested using the test set. The result showed that CNN model still outperforms RNN and LSTM models in terms of accuracy rate, precision, recall and f1-score. Table 5 summarized the result of the test on three models.

Table 5: Testing Result on CNN, RNN and LSTM Models.

	CNN Model	RNN Model	LSTM Model
Test Accuracy	98.46%	97.90%	96.22%
Precision	0.98	0.98	0.96
Recall	0.98	0.98	0.97
F1-score	0.98	0.98	0.97

5 REMARKS

In this paper, the construction of bilingual corpus using parallel data collected from the Web for machines learning systems has been proposed. Using the data after pre-processing, high quality data sets can be prepared for training the CNN model. Using the well-trained model, future parallel data can be selected, classified and added to the bilingual corpus. Training has been conducted to define and evaluate the CNN model. The result showed that CNN outperforms RNN and LSTM in terms of accuracy.

ACKNOWLEDGEMENTS

This work was supported in part by the research grant (No.: RP/ESCA-04/2020) offered by Macao Polytechnic Institute.

REFERENCES

- Lin, H., Tse, R., Tang, S.-K., Chen, Y., Ke, W., Pau, G. (2021). Near-realtime face mask wearing recognition based on deep learning. In *18th IEEE Annual Consumer Communications and Networking Conference (CCNC 2021)*. doi: 10.1109/CCNC49032.2021.9369493
- Chan, K. I., Chan, N. S., Tang, S.-K., Tse, R. (2021). Applying Gamification in Portuguese Learning. In *9th International Conference on Information and Education Technology (ICIET 2021)*, pp.178 – 185. doi: 10.1109/ICIET51873.2021.9419612
- Chan, N. S., Chan, K. I., Tse, R., Tang, S.-K., Pau, G. (2021). ReSPeCT: privacy respecting thermal-based specific person recognition. In *Proc. SPIE 11878, Thirteenth International Conference on Digital Image Processing (ICDIP 2021)*. <https://doi.org/10.1117/12.2599271>
- Tse, R., Mirri, S., Tang, S.-K., Pau, G., Salomoni, P. (2020). Building an Italian-Chinese Parallel Corpus for Machine Translation from the Web, In *6th EAI International Conference on Smart Objects and Technologies for Social Good (GOODTECHS)*. pp. 265-268. doi: 10.1145/3411170.3411258.
- Zin, M.; Racharak, T. Le, N. (2021). Construct-Extract: An Effective Model for Building Bilingual Corpus to Improve English-Myanmar Machine Translation. In *Proceedings of the 13th International Conference on Agents and Artificial Intelligence - Volume 2: ICAART*, ISBN 978-989-758-484-8; ISSN 2184-433X, pages 333-342, doi: 10.5220/0010318903330342
- Cheong, S. T., Xu, J. Liu, Y. "On the design of web crawlers for constructing an efficient Chinese-Portuguese bilingual corpus system," 2018 International Conference on Electronics, Information, and Communication (ICEIC), 2018, pp. 1-4, doi: 10.23919/ELINFOCOM.2018.8330698.
- Cheok, S. M., Hoi, L. M., Tang, S.-K., Tse, R. (2021). Crawling Parallel Data for Bilingual Corpus Using Hybrid Crawling Architecture. In *12th International Conference on Emerging Ubiquitous Systems and Pervasive Networks (EUSPN-2021)*, 198, 122-127. <https://doi.org/10.1016/j.procs.2021.12.218>.
- Sharma, S., Gupta, P. (2015). The anatomy of web crawlers. In *International Conference on Computing, Communication & Automation*. doi:10.1109/cca.2015.7148493.
- Cho, J., Garcia-Molina, H. (2002). Parallel crawlers. In *Eleventh International Conference on World Wide Web (WWW)*. doi:10.1145/511446.511464
- Chakrabarti, S., Berg, M. V., Dom, B. (1999). Focused crawling: A new approach to topic-specific Web resource discovery. *Computer Networks*, 31(11-16), 1623-1640. doi:10.1016/s1389-1286(99)00052-3
- Pappas, N., Katsimpras, G., Stamatatos, E. (2012). An Agent-Based Focused Crawling Framework for Topic- and Genre-Related Web Document Discovery. In *IEEE 24th International Conference on Tools with Artificial Intelligence*. doi:10.1109/ictai.2012.75.
- Li, Y. (2010). Study and implementation on key techniques for an example-based machine translation system. In *Second IITA International Conference on Geoscience and Remote Sensing*. doi:10.1109/iita-grs.2010.5604108.
- Zhao, R., Mao, K. (2018). Fuzzy Bag-of-Words Model for Document Representation. *IEEE Transactions on Fuzzy Systems*, 26(2), 794-804. doi:10.1109/tfuzz.2017.2690222.
- Uchida, S., Yoshikawa, T., Furuhashi, T. (2018). Application of Output Embedding on Word2Vec. In *2018 Joint 10th International Conference on Soft Computing and Intelligent Systems (SCIS) and 19th International Symposium on Advanced Intelligent Systems (ISIS)*. doi:10.1109/scis-isis.2018.00224.
- Ristad, E., Yianilos, P. (1998). Learning string-edit distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(5), 522-532. doi:10.1109/34.682181
- Baker, M., Malmkjær, K. (2001). *Routledge Encyclopedia of Translation Studies*. Psychology Press.
- Zhang, H., Liu, Q. (2002) Model of Chinese Words Rough Segmentation Based on N-Shortest-Paths Method. *Journal of Chinese Information Processing*, vol. 5, pp. 1-7.