Perspectives in Communication, Embedded-Systems and Signal-Processing (PiCES) – An International Journal
ISSN: 2566-932X, Special Issue - 2022

Abstractive Text Summarization using Deep Learning

Akarsh B A
Dept. of Electronics and Communication, Sapthagiri College of Engineering, Bangalore, India

Deepak K
Dept. of Electronics and Communication, Sapthagiri College of Engineering, Bangalore, India

Hemanth H M
Dept. of Electronics and Communication, Sapthagiri College of Engineering, Bangalore, India

L Vishnu Vardhan Reddy
Dept. of Electronics and Communication, Sapthagiri College of Engineering, Bangalore, India

Padmavathi C
Professor, Dept. of Electronics and Communication, Sapthagiri College of Engineering, Bangalore, India

Abstract: In current era, organizing the data has been a very tedious process as there is a lot of data growing every day and storing and managing them is getting tougher, to solve this problem one of the ways is to summarizing the data which in turn will save lot of storage and makes managing data easier. In this paper we have looked into various techniques and methods to summarise the text data using deep learning techniques and come to the conclusion that using Encoder-Decoder Architecture which consists of LSTMs, which is efficient in producing optimum results. This is because LSTMs are good at remembering the long dependencies by overcoming the problem of remembering the context of the previous line in its memory to predict the next sequence. We will first train the model after setting up the model to predict the target sequence. In this training process, the encoder takes an input and processes it and stores the context of that input whereas the Decoder predicts the next word with respect to the previous input. Later a new test input is uploaded to the model to test for which the target is unknown. Attention mechanism is used in this model as it gives attention to the important word in the sentence to understand the context of that sentence and to generate new words from it.

Keywords: LSTMs; Encoder Decoder; Attention

I. INTRODUCTION

Nowadays text is the common format of exchanging information. emails, reports, reviews, messages, etc. are the different sources of text data. There are lakhs of reviews is being given by customers, reporters. The information stored in that reviews is huge and it is complicate to understand the main content. To resolve this problem, we have to extract only main content from the reviews.

Text summarization can be divided into two approaches – Extractive and Abstractive.

The first processing step undergone by the machine is to give an appropriate score to every sentence contained in the datapoints and spots these sentences. In that the peak sentence will be taken as the summary. In this approach there each sentence will be grammatically correct as long as input review given by customers is grammatically correct, so we need not to worry about the grammar of the summary generated. This will be important advantage for this approach.

In the abstractive summarization approach as shown in Fig.1, the review given is first understand by us and hire a new language generation technique for creating the summary, on the use of other words and phrases which are not there in the review. This approach is more like a human way of approach. Whereas we humans will not simply choose the selected sentences for generation of summary. but instead of that we will make use of our own words or expressions to explain the main content of the summary generated or text.

Abstractive base approach has problems and complications. So, it becomes least interested in the research group. In the abstractive-based approach we need to examine the input reviews and new technologies for creating summary. First, Due to the disappointing performance of already existing analysing and generating tools will affect the summarization system poorly. Next, compared to extractive summarization abstractive summarization is harder for the evaluation. Whereas in extractive summarization, we can evaluate the flapping of the output summary with the original standard summary to check the output of the system. But the abstractive summarization will be having different ways to communicate the particular event so, it will not be a good method. Due to the above-mentioned reasons the most of research group focused their efforts on extractive summarization and few of them on abstractive summarization.

II. PROBLEM STATEMENT

In the present generation, there is enormous amount of data generated daily due to digitalization. So, the necessity of understanding the data is important. It is very

crucial to feed the processed data to the model for better efficiency and reduce the time complexity. Customer reviews are usually prolonged and expressive. Understanding these reviews manually is hectic and requires a lot of time. Nowadays people are busy to completely read the reviews, this is where the intelligence of Natural Language Processing and Deep Learning techniques which will help in generating abstractive summaries.

III. RELATED WORK

Yuji Roh et al., [1] Machine learning is used worldwide. It is very necessary to collect large quantities of data. Data gathered is not completely a label data. Data selection can be done as two steps. Firstly, the data generated should be indexed and it should be published for sharing it with others. There are several platforms for sharing the equipped datasets. There are software's for generating data and storing data, as it is the forwarding step in data description. Kaggle, data world is some of sites to visit for datasets. Collaborative systems are used to help us in making the work easy. There are systems that are designed with dataset sharing in mind. they include collaborative analysis, publishing on web.

Vivek Agarwal et al., [2] Data pre-processing is meant for converting raw data into a comprehensive data format. It consists of data cleaning, data integration, data transformation and data reduction. Real world data may be deficient, unnecessary and clamorous. Pre-processing steps include Raw Data, URL removal, hashtag removal, emoticon extraction, stemming, Tokenization, POS tagging, parsing, processed data. The above mentioned are the basic steps we must follow for every dataset. URL and hashtag are used in, as it doesn't give any meaning for the sentences. Regular expressions library works for pattern matching and URL tags detection. The hashtag used in reviews are linking it to the accounts or shortcuts. It is not necessary for the model. We use regular expressions for removal of hashtags. Stemming is the process of reducing the preferred words into the word's stem.

Felix A. Gers et al., [3], alternatives for recurrent neural network, i.e., gated recurrent neural network (GRU) or long short-term memory (LSTM), are considered as the encoder and decoder basic structure. LSTM's advances over traditional RNNs as there is long time gaps for LSTM between the input events as that of recurrent neural network. These networks don't extract information, it stores the information. LSTM gives solution for highly nonlinear tasks. This can be done as LSTMs can learn to precisely measure time intervals at every time step.

Zhenlin Liang et al., [4], Attention mechanism is used widely right now for translation, summarization etc., There are 2 approaches of attention namely global attention and local attention. Local attention takes limited hidden states of encoder and global attention considers all the hidden states. the global attention has a disadvantage of to attend to all the words to source side for each target word. To avoid this problem, we consider local attention in our model.

Hai-Tao Zheng et al., [12], In this paper the attention mechanism is used to generate news comments. The generation of news comments is a latest, challenging and not so well-studied task of NLG. Unlike the other NLG tasks, the generation of news comments requires the contextual correlation between comments and the respective new. Also, with that, we should generate different kinds of comments to diversify it, because usually people with different mindset will have different types of opinions on a particular topic in reality. Gated attention neural network model must be used to provide the contextual relevance in generate news comments. To solve the problem of contextual relevance, we also need to insert the gated attention mechanism to use news context self-adaptively and selectively. The contextual information from the news must be utilized properly to concentrate different type of words to the different parts of the title context.

IV. OBJECTIVES AND METHODS

A. Objectives

The five main objectives of this project are:

1) To choose an appropriate dataset.
2) To pre-process the data.
3) To build an abstractive model.
4) To train, test and predict the model on different parameters.
5) To translate into required language.

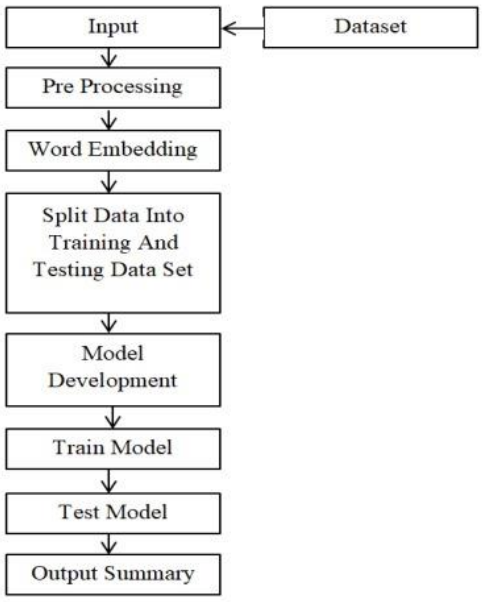

Fig 1. Steps for Abstractive summarization model

The above block diagram shows the steps for building an abstractive text summarization model.

B. Methods

a) Data Description and Pre-processing

The dataset called "Amazon Fine Food Reviews" from Kaggle is selected for the summarization purpose. A file called "Reviews.csv" is used to summarize the reviews. The Review file consist of 5,68,454 reviews in a column called Text, for each of which consists of columns namely: Id, Product Id, User Id, Profile Name, Helpfulness Number, Helpfulness, Score, Time and Summary. Out of which we will be using Text and Summary Columns.

Data pre-processing is process of simply transforming raw data into understandable data format. It consists of data cleaning, data integration, data transformation, data reduction, and data discretization. [5] Real world data is sometimes incomplete, inconsistent, redundant and noisy. Pre-processing steps include Raw Data, URL removal, hashtag removal, contraction mapping, removing text inside parenthesis, removing stop and short words, removing punctuations and special characteristics, [6] Tokenization, stemming, POS tagging, parsing, processed data. The above mentioned are the basic steps we must follow for every dataset.

URL and hashtag are used in tweet do not play essential role in bringing the sentiment or it's doesn't give any meaning for the sentences. Pattern matching and URL detection can be done with the help of library called Regular expression. The hashtag used in reviews are linking it to the accounts or shortcuts. It is not necessary for the model. We use regular expression module for removal of hashtags. In contraction mapping We defined all contraction words and convert them into text, using regular expression module we removed the text inside the parenthesis, and eliminated the punctuations and special characteristics. Then removed short and stop words.

Parts of Speech (POS) Tagging is also called as word category disambiguation. It is a technique for correcting a word in corpus with respect to a fragment of speech. With the help of NLTK library, we have produced the POS tags for its respective tokens earlier generated [8]. NLTK consists of Stop words, start words of every language. Checking it after the generation of token and then we must tag them. The final step is to perform the semantic analysis which draws out the meaning of the input. To do this we have to define a CFG and produce a parse tree from it. Due to this reason, we have used Penn Treebank corpus. If a token is stop-word, then we can add a subtree "(STOP (stop word))" to the checked tree [7]. It is not necessary for these subtrees to be traversed to understand the sentiment of the review. Prior to the building of checked trees, the named entities which were produced before for the processed tweets. These are the steps of pre-processing. Now the data corpus is ready for model building. By this, our paper exhibits that it is possible to produce the datasets by rejecting the unwanted tweets from bots and focusing on the tweets written by humans. By this it reduces the processing difficulty and gives a efficient dataset.

b) Sequence-to-Sequence model

Sequence-to-Sequence model are used when dealing with the sequence data. The model converts a sequence data from one domain to a sequence data in another domain. The models basically consist of two elements which are one encoder and one decoder network as shown in the Fig. 2. The encoder network converts each of the it into a respective hidden vector consisting of it and its context. decoder does reverse process of encoder, converting the hidden vector into an item, with the help of previous output as a input context. This kind of models can be done by using recurrent neural network (RNN) or Long Short-Term Memory (LSTM) or Gated recurrent unit (GRU) to avoid the problem of vanishing gradient.

This model tells the importance of words relatively to distinguish between content words and stop words. To overcome some modelling issues, we use deep convolutional encoder for input sequence [13]. This kind of architecture helps to enhance on the bag-of-words model by the allowing locally happening interactions between words during the process of encoding the input. Even though convolutional encoder has high capacity, it is required to generate single representation for the whole input sequence. We have combined the probabilistic model and a generation algorithm which can produce precise abstractive summaries. We would further improve the grammar of the summaries

Encoder is a set of many recurrent units which takes a single element of the input sequence, collects the information for that element and propagates it forward. Similarly, decoder is also a set of many recurrent units where each predicts an output at each given time t. the model has several hidden states in between encoder and decoder. The hidden states of encoder are computed using the formula shown below.

$$h_t = f(W^{(hh)} h_{t-1} + W^{(hx)} x_t)$$

The hidden states of decoder are computed using the following formula shown below.

$$h_t = f(W^{(hh)} h_{t-1})$$

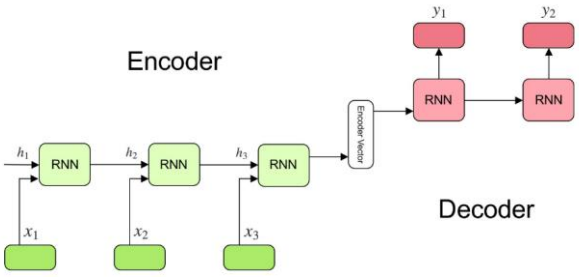

Fig 2. Sequence to Sequence model.

Recurrent neural network is class of neural network that helps in modelling sequence data [9]. RNN coverts independent activation function by giving the same weights and biases to each and every layer, so decreasing

the complexity of increasing parameters and remembering each of the previous outputs by giving each the output as input to the next hidden layer. RNN memorizes every information through time.it is best kind of time series prediction because of the feature to memorize previous inputs as well [10].

c) LSTM

There has been a problem of sequence prediction for a long time. The encoder consists of a stacked LSTMs, while the decoder consists of a Uni-directional LSTMs where the size of hidden state is same as the encoder structure and a layer of attention mechanism is added over the source unseen states and a soft-max layer is also added on the target vocabulary which will help to generate new words. Different types of recurrent neural networks like gated recurrent neural network (GRU) or long short-term memory (LSTM), are liked as encoder and decoder components as shown in Fig. 3. the keywords are captured using the feature-rich encoder. This is considered as toughest problem to solve in the data science industry. Some of the problems are predicting the future sales of a product with the help of previous months data. Long Short-Term Memory networks has been more effective solution for this problem. In comparison to feed-forwarding neural networks and RNN, LSTMs is better at this as it can remember the patterns for a longer time. RNNs doesn't remember a certain important word, it instead changes the whole sentence by applying to a function [15]. Whereas LSTMs can capture or forget things so it makes few modifications to the data by addition or multiplication. These information in the LSTMs flows through a process called cell states. An LSTM consists of blocks of memory called cells which stores the data.

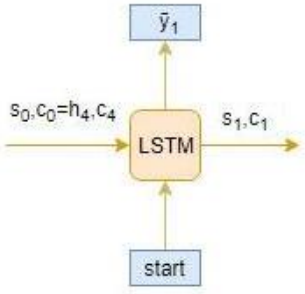

Fig 3. Single LSTM Network.

d) Attention Mechanism

The problem with encoder -decoder approach is that it needs to compress all the necessary sentences into a defined length of vector. It is hard to understand long sequences. To overcome this issue attention mechanism is used.

Attention mechanism predicts the words by looking to a specific word in a sequence. So, there is no need of looking to the whole sequence instead we can concentrate more on specific words in the sequence which give the output sequence.

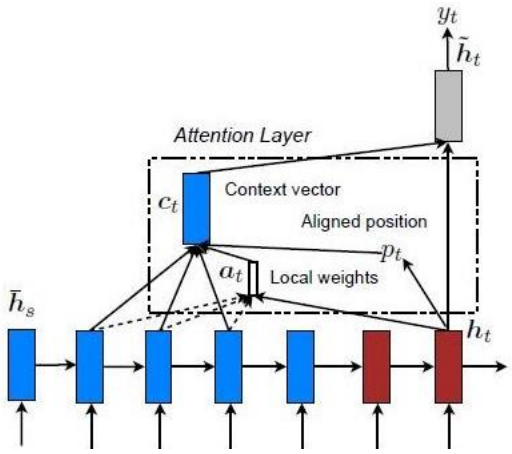

Fig 4. Attention Mechanism.

There are two types of attention namely local attention and global attention as shown in the above Fig. 4. Working of attention mechanism comprises of four steps. They include calculation of alignment scores, normalizing the alignment scores with the activation functions like SoftMax functions, next step is to calculate the context vector of each time step. The context vectors are concatenated to form attended hidden vector. Lastly the attended hidden vector is fed into dense layer to produce target vector.

There are many ways for machine translation, mainly two categories i.e., Rule-Based and Empirical-Based Machine Translating systems. The Rule-Based Translation which is further classified into Direct, Transfer and Interlingua. The Empirical-Based Machine Translation System is further classified into Statistical and Example based machine learning system [14]. By taking the advantages of these both Hybrid-Based Translation is confirmed to have better efficiency in the Machine Translation Systems [16].

e) Model Building

The complete data can be divided as train data and test data in a ratio of 90:10 using a function from sklearn library. A neural network comprising of input layers, hidden layers and output layers are added on the model. A 3 stacked Long Short-Term Memory can be built for the architecture of encoder-decoder. The model consists of attention layer and dense layer using an activation function called SoftMax. The output of the attention layer which will take the output of the encoder is given to the decoder. For compilation of the model rmsprop optimizer is used. The loss function of the model can be calculated by Sparse Categorical cross entropy. Which will convert the integer sequence to one hot vector.

f) Model Deployment

One of the biggest concerns of training a model is overfitting. To avoid this problem a module called earlystopping is used. The training of the model can be

stopped by the earlystopping module at the correct time by observing the user-specified metric. The training will be stopped by the model when the evidence loss is raised for the two successive epochs.

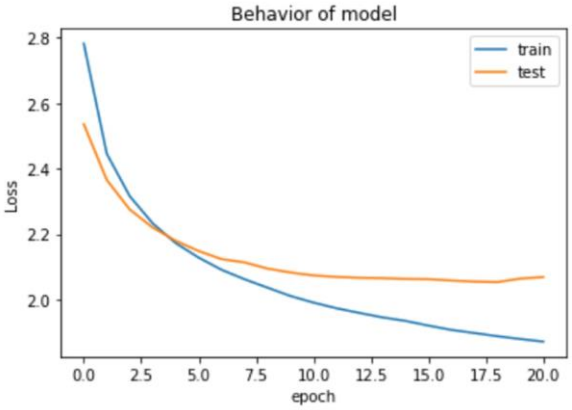

Fig 5. Behaviour Model.

The above Fig. 5 is the graph plotted against loss on y-axis and epoch on x-axis for train and test data. The loss during training is train loss and loss during testing is validation loss. As shown in the figure, the training is stopped when the validation loss starts to increase.

C. Advantages And Application

- It reduces the reading time of the user.
- Removes irrelevant words from text.
- Predicted summaries minimizes the required storage.
- It reduces the time complexity of the model.
- These models are implemented in softwares like Grammerly, Linguix, ProWritingAid and etc.,
- These are also implemented in NewsLetters, Social media marketing and etc.,

V. RESULTS AND DISCUSSION

The below Fig. 6 shows the model summary. This model comprises of input layer, embedding layer, set of LSTMs layers, attention layer and dense layer.

Fig. 7 shows the final output generated by this model. The first point is the processed input given by user. The Original summary is also provided for training purpose. The predicted summary is generated by the model and then translated to required language.

VI. CONCLUSION

In this paper we have proposed attention based abstractive summarization. Since we are using attention layer our model consumes less time. We have approached a better way of sequence-to-sequence modelling using 3 stacked LSTMs as it comprises of three hidden Long Short-Term Memory layers for each hidden layer accommodate various memory cells. Stacked LSTMs will help us reach greater model complexity. The statistical idea behind our model is calculating loss, using Sparse Categorical Cross Entropy whereas the advantage is that it saves time in memory and computation. During the training stage, the model training was interrupted due to Early stopping as the validation loss started to increase consecutively.

```
Model: "model"
__________________________________________________________________________________________________
Layer (type)                    Output Shape         Param #     Connected to
==================================================================================================
input_1 (InputLayer)            [(None, 30)]         0
__________________________________________________________________________________________________
embedding (Embedding)           (None, 30, 100)      1360500     input_1[0][0]
__________________________________________________________________________________________________
lstm (LSTM)                     [(None, 30, 300), (N 481200      embedding[0][0]
__________________________________________________________________________________________________
input_2 (InputLayer)            [(None, None)]       0
__________________________________________________________________________________________________
lstm_1 (LSTM)                   [(None, 30, 300), (N 721200      lstm[0][0]
__________________________________________________________________________________________________
embedding_1 (Embedding)         (None, None, 100)    365100      input_2[0][0]
__________________________________________________________________________________________________
lstm_2 (LSTM)                   [(None, 30, 300), (N 721200      lstm_1[0][0]
__________________________________________________________________________________________________
lstm_2 (LSTM)                   [(None, 30, 300), (N 721200      lstm_1[0][0]
__________________________________________________________________________________________________
lstm_3 (LSTM)                   [(None, None, 300),  481200      embedding_1[0][0]
                                                                 lstm_2[0][1]
                                                                 lstm_2[0][2]
__________________________________________________________________________________________________
attention_layer (AttentionLayer ((None, None, 300),  180300      lstm_2[0][0]
                                                                 lstm_3[0][0]
__________________________________________________________________________________________________
concat_layer (Concatenate)      (None, None, 600)    0           lstm_3[0][0]
                                                                 attention_layer[0][0]
__________________________________________________________________________________________________
time_distributed (TimeDistribut (None, None, 3651)   2194251     concat_layer[0][0]
==================================================================================================
Total params: 6,504,951
Trainable params: 6,504,951
Non-trainable params: 0
```

Fig 6. Model summary

```
Review: delicious guilt free snack always handy convenient last long time also
get corn mixed veggies also wonderful get whole foods price
Original summary: love
Predicted summary:  delicious
Translated summary:  ರುಚಿಕರವಾದ


Review: usually drink click water ice blender would think gone favorite coffee
shop got frozen coffee drink use blender never lumps hope click lovers try way
enjoy much
Original summary: love this
Predicted summary:  my favorite drink
Translated summary:  ನನ್ನ ನೆಚ್ಚಿನ ಪಾನೀಯ
```

Fig 7. Final Output

A. Future work

- We can implement this model using Bi-directional LSTMs instead of stacked LSTMs.
- For decoding the text sequence, we have used greedy approach, instead we can try implementing beam search strategy for more efficiency.
- We can use pointer-generator networks as an added feature for attention model.

REFERENCES

[1] Yuji Roh, Geon Heo, Steven Euijong Whang, Senior Member,"A Survey on Data Collection for Machine Learning A Big Data - AI Integration Perspective", IEEE, vol 3, pp 211-223, Aug 2019.

[2] Vivek Agarwal "Research on Data Pre-processing and Categorization Technique for Smartphone Review Analysis" International Journal of Computer, vol 13, pp 1-5, Dec 2015.

[3] Felix A. Gers, Nicol N. Schraudolph, J¨urgen Schmidhuber, "Learning Precise Timing with LSTM Recurrent Networks ", Journal of Machine Learning Research, vol 3, pp 115-143, April 2002.

[4] Zhenlin Liang, Chengwei Huang, "Speech Emotion Classification Using Attention Based LSTM", IEEE/ACM Transactions on Audio, Speech, And Language Processing, vol. 27, pp 11, Nov 2019

[5] Bhardwaj, A. Deshpande, A. J. Elmore, D. Karger, S. Madden, A. Parameswaran, H. Subramanyam, E. Wu, and R. Zhang, "Collaborative data analytics with datahub," PVLDB, vol.8, pp 1916–1919, Aug 2015.

[6] Timothy E. Ohanekwu, Ezeife "A Token-Based Data Cleaning Technique for Data Warehouse Systems" journal of the Association for Computing Machinery (ACM) vol 1, pp 12-22, June 2003.

[7] Fakhitah Ridzuan, Wan Mohd Nazmee Wan Zainon "A Review on Data Cleansing Methods for Big Data "The Fifth Information Systems International Conference vol 4, pp 12-23, Mar 2019.

[8] Wei Jianping, "Research on Data Preprocessing in Supermarket Customers Data Mining", IEEE Conference Information Engineering and Computer Science (ICIECS), 2010 2nd International Conference, pp 1 – 4, Dec 2010.

[9] Ramesh Nallapati, Bowen Zhou, Cicero dos, Çaglar Gulçehre, Bing Xiang "Abstractive Text Summarization using Sequence-to-sequence RNNs and Beyond", Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning (CoNLL), pp 280–290, Berlin, Germany, Aug 2016.

[10] Mike Schuster and Kuldip K. Paliwal, "Bidirectional Recurrent Neural Networks". IEEE Transactions on Signal Processing, vol. 45, pp 11, Nov 1997.

[11] Minh-Thang Luong, Hieu Pham, Christopher D. Manning, "Effective Approaches to Attention-based Neural Machine Translation", Journal arXiv preprint, vol 2, pp 22-27, Aug 2015.

[12] Hai-Tao Zheng, Wei Wang, Wang Chen, And Arun Kumar Sangaiah, "Automatic Generation of News Comments Based on Gated Attention Neural Networks", vol 6, pp 1-4, Jan 2018.

[13] Alexander M. Rush, Sumit Chopra, Jason Weston A Neural Attention Model for Sentence Summarization Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pp 379–389, Sep 2015.

[14] Simarn N. Maniyar, Sonali B. Kulkarni, Pratibha R. Bhise, "Systematic Review on Techniques of Machine Translation for Indian Languages", International Journal of Emerging Trends & Technology in Computer Science vol 9, pp 15-19, Sep-Oct 2020.

[15] Novriyanto Napu, Rifal Hasan, "Translation Problems Analysis of Students" International Journal of Linguistics, Literature and Translation (IJLLT) vol 2, Sep 2019.

[16] Alejandra Hurtado de Mendoza, "The Problem of Translation in Cross-Cultural Research on Emotion Concepts", International Journal for Dialogical Science vol 3, pp 241-248, Feb 2008.