Language Trainer

Yuanlai He, Angel Salazar, Wei Yan

***Abstract*—This paper explores the critical role of deep learning in various sectors, emphasizing its impact on learning environments and industry-specific applications. The article introduces an innovative approach to utilizing Neural network technology and deep learning for optimizing new language learning through mobile devices. The work relies primarily on the smartphone's functions to translate text to speech, dictate text to be compared, and provide feedback and areas of improvement to the user. The paper outlines this innovative software's architectural framework and database schema, highlighting user interface integration, topic content data, user record history, and analysis. Furthermore, it discusses the user authentication, topic hub, progress summary, account configuration, and analysis stage integration aspects of the associated iOS app. The conclusion addresses the successful implementation of the software and outlines future work to keep improving the user's experience.**

***Index Terms*—Siamese neural networks, deep learning, iOS app development, new language learning, text-to-speech.**

I. Introduction

Deep learning transforms our lives and industry processes. It refers to machine learning technologies for learning and utilizing artificial neural networks, and it has successfully been applied to natural language processing (NLP). NLP comprises five major tasks: classification, matching, translation, structured prediction, and sequential decision process. One outstanding application for translation is automatic speech recognition [1].

Machine translation is one of the applications of NLP. It involves using techniques to translate documents from one language to another. In translation tasks, English is almost always either the input or output language, with the other end usually being a major European or Eastern Asian language. There are thousands of languages spoken worldwide, meaning that a system capable of collecting and validating data in other languages will be a huge contribution to NLP and humankind [2].

Measuring Semantic Textual Similarity (STS) calculates the similarity between a pair of texts and is very important for many NLP applications. For example, an accurate STS component is the key to success for question-answering systems since the questions with similar meanings can be answered similarly. One recent outstanding STS method is Siamese neural networks, they are helpful among tasks that involve finding similarity or a relationship between two comparable things. They have been proven successful in tasks like signature verification, face verification, image similarity, and sentence similarity [3].

The use of chatbots, which are part of the development of artificial intelligence, in English language learning has positive and negative perceptions about its natural language processing utility. Some factors that favour the behavioral intention of students to use Chatbots are the ease of use, the content of one's choice, confidence as a byproduct, the absence of time, and cost and place contingencies. On the other hand, a sense of robotic interaction, emotionlessness, and lack of flow in conversation are factors that demotivate students' intention to use Chatbots [4].

In this paper, the use of deep learning is focused on improving the English learner's level. Mobile device tools like text-to-speech and text dictation features will be used. The information display and interaction with the user will be carried with a friendly, easy-to-use user interface (UI) in an iOS app. Furthermore, artificial neural networks will help evaluate the user's performance, improving and innovating the learning experience.

This paper is divided into sections. The first one is the introduction, where the importance of deep learning in speech processing tasks has a quick approach and explanation. The Siamese neural network explanation is in section II. The training and testing are in section III. Then, the main architecture of the system is presented in section IV. section V goes through details about the API and UI. Conclusions are then drawn in Section VI.

II. Siamese Network Model

Siamese networks are a special type of neural network which have two identical subnetworks merged in the output. Unlike the classification task, it learns what makes the two inputs the same, such as in face recognition, signature verification, and text similarity recognition for our task. Fig. 1 illustrates an example of Siamese network architecture.

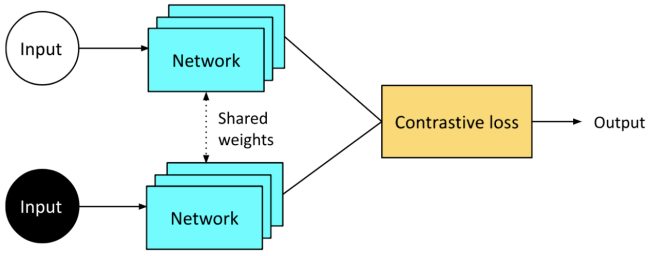

Fig. 1. Example of a Siamese network architecture.
Source: Adapted from [5].

In our task, for each subnetwork, an embedding layer and a bidirectional long short-term memory (BiLSTM) layer are used to extract the feature of the input sentence. The feature vectors are compared by the cosine-based similarity. The cosine similarity is a measure of similarity between two non-zero vectors of an inner product space that measures the cosine of the angle between them. Fig. 2 illustrates our

Siamese network architecture. This algorithm is the key to calculating the similarity between the script and the recognized text.

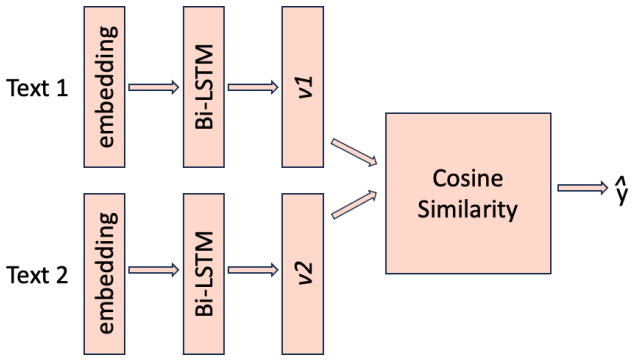

Fig. 2. Our Siamese network architecture.

A. Embedding Layer

The first layer of this model is the embedding layer. It converts input vocabulary indices into dense vectors of a fixed size (embedding_dim), which is crucial for processing textual data. Unlike simpler representations like one-hot encoding, embeddings can capture richer semantic information, allowing the model to learn the best representation of words in the context of the specific task.

B. BiLSTM Layer

The second layer is a bidirectional Long Short-Term Memory (LSTM) network that processes sequential data. The bidirectional nature allows the network to capture information from both past and future states.

C. Mean and Normalization

After processing through BiLSTM, the mean of the output sequence is computed, which effectively averages the features across all words/time steps for each sample in the batch. Followed by L2 normalization, this step is essential for preparing the data for similarity comparison. It adjusts the components of a vector so that the vector has a length (or norm) of 1 but retains its direction in the vector space, which maintains scale consistency, improves numerical stability, and facilitates the calculation of cosine similarity.

D. Loss Function

For this specific siamese network, we have a triplet loss function ensuring that embeddings of similar items (positive pairs) are closer together than embeddings of dissimilar items (negative pairs) by a certain margin.

$$scoresMatrix = v_1 \times v_2^T$$

v_1 and v_2 are the output vectors of a batch of the two branches of the Siamese Network. The score matrix's diagonal represents the positive pairs' similarities (cosine similarities of matching embeddings). The non-diagonal elements represent the similarities of the negative pairs.

$$Cost1 = max(-cos(A,P) + mean\_neg + \alpha, 0)$$
$$Cost2 = max(-cos(A,P) + closest\_neg + \alpha, 0)$$

where cos(A, P) is the cosine similarity of the positive pair, mean_neg is the mean negative similarity, and closest_neg is the highest similarity among the negative pairs.

$$Triplet\ Loss = \frac{1}{N}\sum(Cost1 + Cost2)$$

where N is the batch size.

III. Model Training

A. Data overview

The dataset used in this project is Quora's first public dataset regarding the problem of recognizing duplicate questions[6]. It is based on actual data from Quora, helping us train and text the semantic equivalence model.

Our model also focuses on recognizing the similarities between the original and recognized texts. For instance, “How can I be a good geologist?” and “What should I do to be a great geologist?” are the same questions. The total amount of samples in this dataset is over 400,000. Each sample has a duplicate question pair, ID, and a binary Integer of whether the question pair is duplicate.

B. Data preprocessing

Data preprocessing involves several stages, including splitting the dataset, tokenizing and padding the text sequences.

The dataset is split into training and testing sets, with 394100 and 10240 entries, respectively.

We extract duplicate question pairs from the training dataset for the Siamese network. Then, separate arrays for the first and second questions in training and testing datasets are created.

Then, build the vocabulary, tokenizing each question using NLTK's word_tokenize. NLTK is a natural language tool kit library in Python. A vocabulary dictionary is built by mapping each unique word to a numeric index. Next, the vocabulary index can be used to convert the tokenized questions into a sequence of numbers.

Then, after determining the maximum length of the questions in the training set, each question was padded with 1 to ensure that all sequences were of the same length.

Finally, the training data is further divided into an 80% training set and a 20% validation set.

C. Hyperparameter

The hyperparameter of our model training is shown in the table 1. The threshold is used to determine whether a similarity can indicate if the two inputs are duplicates. The learning rate of 0.01 and epochs of 10 are good enough for

training this model.

Table 1. Hyperparameters

Hyperparameter	Value
Learning rate	0.01
Epochs	10
Optimizer	Adam
Loss function	*Triplet Loss*
Embedding dimension	128
LSTM hidden size	128
Batch size	256
Threshold	0.7

E. Testing Result

The overall test accuracy is 70.29%. As shown in Fig. 3, the confusion matrix indicates that the model can recognize the similarities of texts effectively.

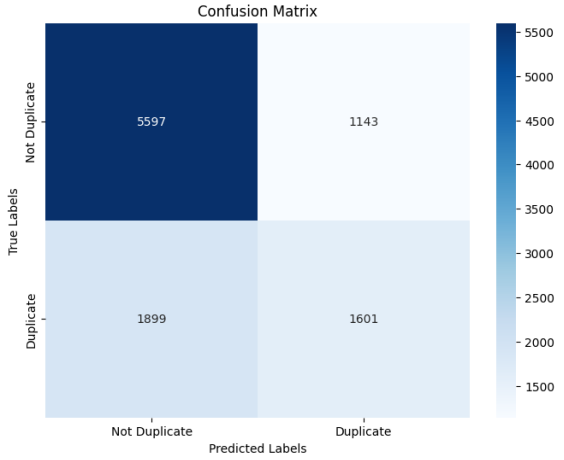

Fig. 3. Confusion Matrix

F. Costum Test

We also test it in our own examples. For instance, we test it with "*Cats never fail to fascinate human beings.*" and "*Cards never fail to fascinate human beings.*" and we get a similarity of 74%. we also test it with "*They can be friendly and affectionate towards humans, but they lead mysterious lives of their own as well.*" and "*They can be friendly and a fake Sinead towards requirements, but the lead mysterious lives of their own as well.*" and we get a similarity of 71%. These results indicate that the model can recognize the similarity of two natural texts relatively.

IV. Main architecture

The architectural framework of this project comprises several integral components, each serving a distinct purpose in orchestrating its functionality.

A. iOS application with Swift UI

Swift programming software is fully utilized in this project to develop a UI on iOS that allows the user to navigate through different pages like login, sign up, account settings, script display, pitch feedback and history of records. Since it is currently only a minimum viable product, we have only developed the mobile app on the iOS platform.

B. Amazon Elastic Compute Cloud

The Amazon Elastic Compute Cloud (Amazon EC2) allows us to deploy a flask API that contains our learning model. It communicates with the smartphone through the Internet, as shown in Fig. 4.

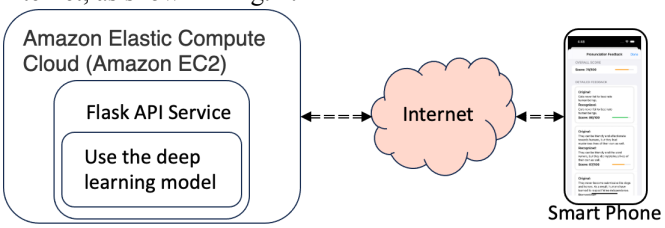

Fig. 4. Amazon EC2 and smartphone connection diagram.

V. App Frontend

The frontend is an iOS app developed in Swift. It can display an English script and a template audio speaking by a native speaker or AI. The user can listen to the audio of the script by clicking a button. Then, the user can record his/her voice, and the audio will be converted to text by the iOS speech recognition API. Then, the server will compare the recognized text with the original text and provide a score to the user based on the similarity of the sentences. Fig 5. shows the different functions and screen flow that can be displayed.

Fig. 5. App UI flowchart.

VI. Conclusion and Future Works

This project successfully built a language learning trainer by integrating speech recognition, text-to-speech, and artificial neural networks into an iOS app. The UI can display different topics to pick scripts from, provide a score, and manage the previous recordings from the user.

In the next stage, the focus will be on adding a transactional database. The dataset will include the user profile data, user record analysis model output, and user record history. This is to manage the large amount of data from all users and customize their experience with the choice of topics.

References

[1] Hang Li, “Deep learning for natural language processing: advantages and challenges”, National Science Review, Volume 5, Issue 1, January 2018, Pages 24–26, https://doi.org/10.1093/nsr/nwx110

[2] D. W. Otter, J. R. Medina and J. K. Kalita, "A Survey of the Usages of Deep Learning for Natural Language Processing," in IEEE Transactions on Neural Networks and Learning Systems, vol. 32, no. 2, pp. 604-624, Feb. 2021, doi: 10.1109/TNNLS.2020.2979670.

[3] T. Ranasinghe, C. Orasan, R. Mitkok, “Semantic Textual Similarity with Siamese Neural Networks”, Proceedings of Recent Advances in Natural Language Processing, pp. 1004-1011, Sep 2-4, 2019, https://doi.org/10.26615/978-954-452-056-4_116

[4] N. Annamalai, R. A. Rashid, U. M. Hashmi, M. Mohamed, M. H. Alqaryouti, A. E. Sadeq, “Using chatbots for English language learning in higher education”, Computers and Education: Artificial Intelligence, Volume 5, 2023, 100153, ISSN 2666-920X, https://doi.org/10.1016/j.caeai.2023.100153.

[5] Gustavo H. de Rosa, João Paulo Papa, Chapter 7 - Learning to weight similarity measures with Siamese networks: a case study on optimum-path forest, Editor(s): Alexandre Xavier Falcão, João Paulo Papa, Optimum-Path Forest, Academic Press, 2022, Pages 155-173, ISBN 9780128226889, https://doi.org/10.1016/B978-0-12-822688-9.00015-3.

[6] Kaggle. (2016). Quora Question Pairs dataset [Data set]. Kaggle. https://www.kaggle.com/datasets/quora/question-pairs-dataset/data