Design and Implementation of Consecutive Interpreting System Based on Transformer NMT Model

Lin-gen Li*, Shuo LI, Wan-qing CUI and Kai WEI
Northeastern University, No. 3-11, Wenhua Road, Heping District, Shenyang, P. R. China
*Corresponding author

Keywords: Consecutive interpretation, Machine translation, Transformer, Attention mechanism, Natural language processing.

Abstract. The traditional machine translation system with push-to-talk mode is not suitable for the processing of long-time oral translation. This paper proposed a consecutive interpreting system, solving the problem of long-time continuous listening by using pipeline work mode. In this mode, audio sampling is always on during the whole speech. In usage scenarios, audiences of the speeches or lectures can see bilingual subtitles on the projection or on their own device, and this system will keep translating while listens to the speaker. The speech-to-text module is based on the speech recognition model of Baidu open platform, and the translation is based on the Transformer NMT model proposed by Google. The average translation delay time of our system is only about 0.8s in our delay test. This system can play the role of the interpreter in conferences or lectures where translation precision requirement is not high.

Introduction

Interpreters play an important role in cross-language communication. There are mainly two modes of interpretation, simultaneous interpretation (SI) and consecutive interpretation (CI). This paper mainly discusses the CI. CI is often used in scenarios such as speeches, lectures and meetings, where interpreters are required to continuously listen to the speaker and the speaker may wait for the interpreter to translate and tell listeners about the translation before continuing to speak.

In terms of delay of translating, a machine translation system does have an advantage over a human interpreter. It takes shorter time to provide the translation of a single sentence though its accuracy is not good enough. However, traditional machine translation systems mostly work in push-to-talk (PTT) modes. A PTT mode translation system required users to push a “button” before they speak and release the “button” when they finish speaking, which means it can’t listen to the speaker’s voice while translating his or her words synchronously. This is extremely undesirable for speakers in the actual application scenarios where needs the CI.

A consecutive interpreting system based on attention mechanism (Transformer NMT model) [1] aimed at the solving problems above is proposed in this paper. This system does not require the speaker to deliberately wait for the completion of its translation, which means the speech process will be smoother. When it runs, it keeps to listening to the speaker, recognizing the voice and translating recognized result to the target language while showing recognized text and translated result to audiences of the speaker.

Consecutive interpreting is a one of interpreting modes in which a speaker delivers information in units of sentences or paragraphs. Consecutive interpreters are used for many occasions, such as speeches, congratulations, lectures, meetings, and press conferences [2]. In these scenarios, the interpreter is required to continuously listen to the speaker, and the speaker will wait for the interpreter to translate and then tell the listener about the translated content before speaking.

Machine translation technology was firstly proposed by Google in the paper “Phrase-Based Statistical Machine Translation” (PBMT) [3]. As the development of deep learning, Google's NMT model based on deep learning [4] used to be popular for translation systems. The feasibility of this
model is high, and after adding the attention mechanism, the translation result become more accurate. The accuracy of the newly proposed Transformer model of Google is also acceptable while the model has lower complexity, which makes it easier to be trained [1]. Therefore, the Transformer model is used in this system as a support for machine translation.

**Status of Machine Translation or Interpretation Systems and Devices**

There are some portable translating devices in the market. They can be divided into two kinds: wireless translation headset and translation machine.

For the wireless translation headset, it needs to work with the mobile phone App. It only supports Android phones. In terms of technology implementation, this headset only plays the role of audio sampling. The translation function is still performing by the mobile App with connection to the cloud server, which is dependent on the network. And it is only used for chat and short sentence translation.

In addition to translating headphones, translation machine is also a common form of translation device. Some translation machine will be equipped with a small screen that displays recognized and translated text, while others have no screen and can only perform dialog translation. However, none of the translation forms of these devices is applicable for consecutive speech.

The usage scenes of above-mentioned translation machine and the translation headset are daily conversations, the travel to a foreign country and so on. This kind of conversational communication does not applicable for needs of consecutive translation scenarios such as conferences and lectures.

The similarity between the system we proposed in this paper and the above-mentioned translation machine is that we all use deep learning based on translation models. The difference is that this system circumvents the push-to-talk or half-duplex mode of the above-mentioned translation systems and can perform continuous listening and translation to meet the requirement of consecutive interpretation.

**The System Framework**

A complete machine interpreting process includes two main steps: voice recognition and text translation. According to the characteristics of machine interpreting, we conclude that the flow chart of a single machine interpreter as shown in Figure 1.

![Flow chart of interpretation work for single sentence.](image)

Based on the above process model, we should design a system with interaction, sound sampling, speech recognition and machine translation functions. Now, the functions are divided into various functional blocks in the system. We have obtained a cross-functional flow chart for a single machine interpretation task as shown in Figure 2.
As shown in the above figure, an interpreting system that satisfies the requirements should be divided into the four modules (functional blocks). Among them, the interaction functional block is used to make the system controllable for the user (speaker). The audio sampling functional block is used for recording and preprocessing audio (such as noise reduction, blank segments removal, etc.). The voice recognition module is used to convert speech audio into readable text, meanwhile do some pretreatment (such as text trim, punctuation optimization, etc.). And the machine translation module do machine translation based on the Transformer NMT model on the recognized result from the voice recognition module.

Consecutive interpreting requires the system to continuously listen to the speaker's speech. Since the human brain cannot be multi-tasking, the human interpreter needs the speaker to stop waiting for him or her to finish the translation. For a machine interpreter system, it has a better multi-task ability than humans, which makes it possible to perform translations and output result while listening.

As shown in Figure 3, we proposed a working flow mode for this situation. This kind of operation mode is similar to the pipeline mode. Pipeline operation is a kind of management mode of project organization implements. It is a kind of method that the workers of the fixed organization successively work in several construction environments with the same work content. This organization method greatly improves work efficiency and is widely used in microprocessors. As we can see, the audio sampling work in this mode covers the total period of time. And data buffers are
necessary in this mode in order to solve potential problems caused in situation that others functional blocks are occupied when the audio sampling functional block finish the listening of a segment. In the pipeline mode, the interpretation system will be immediately do the translation work while listen to the next segment after listening to a segment of speech, without the need to pause speaker’s speech. We use the multi thread function of Java to implement this mode in our system.

Implementation of the System

To implement such a system, we need to build local terminal devices and cloud servers (we mainly talk about the software on the server). The interactive functions and audio sampling functions in the framework should be performed by the local terminal device. The voice recognition function and the machine translation function are supposed to be performed by the server. In this section, we discussed about details about the implementation of two part of the system that are mentioned above.

The Local Terminal Device

The local terminal device needs to be responsible for user interaction and voice sampling functions. We use the Raspberry Pi 3 as the basic hardware platform, with Android Things embedded operating system installed on it. And the interactive user interface of the terminal is built as an Android Things App set as “IoT launcher”, making the app starts every time the device startups. For the sound sampling module, we use a single-ear Bluetooth headset with a microphone (model QCY MINI-1) and we add the code for actively binding this Bluetooth headset device to the terminal to make it automatically connect to the headset when possible.

The pipeline work mode makes the local terminal to continuously sample the user's speech. In the pipeline mode, we use Voice Activity Detection (VAD) algorithm to determine the user's speaking gap. When there is a gap, the currently recognized audio will be saved and a voice recognition will start in a new thread, and the audio sampling module continues to work. In the thread working for voice recognition, the saved audio will be transmitted to the cloud server with STT model. And the cloud server returns the recognition result. After receiving the recognition result, the local terminal transmits the recognition result to the cloud server responsible for translation. After the translation result return, the local terminal will transmit the bilingual sentence pair to the subtitle display program on the desktop of the computer, who displays the subtitles on the projection. In order to make every communication real-time, the communication protocol between the local terminal, server and the subtitle display program are all WebSocket [5]. WebSocket is a protocol that allow two device exchange real-time data with each other on the Internet [6], which satisfies the requirement of the communication of the system.

The Server

The server we set up mainly contains the Transformer NMT model and controller that records the recognition and translation results at the same time for extra usage (e.g. push every result to subtitle display program on desktop). The NMT model we trained only supports English-Chinese translation. For the STT (speech-to-text, voice recognition), we use the Baidu open platform DNN STT model. The reasons why we use the Transformer model are its unique self-attention mechanism, simpler structure and relatively higher performance. The complexity list is shown as Table. 1.

Table 1. Model complexity compare. \( n \) is the sequence length, \( d \) is the representation dimension, \( k \) is the kernel size of convolutions and \( r \) the size of the neighborhood in restricted self-attention. [1]

<table>
<thead>
<tr>
<th>Type of layer</th>
<th>Complexity of the layer</th>
<th>Sequential Operation</th>
<th>Max Path Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>Self-Attention</td>
<td>( O(n^2 \cdot d) )</td>
<td>( O(1) )</td>
<td>( O(1) )</td>
</tr>
<tr>
<td>RNN</td>
<td>( O(n \cdot d^2) )</td>
<td>( O(n) )</td>
<td>( O(n) )</td>
</tr>
<tr>
<td>CNN</td>
<td>( O(k \cdot n \cdot d^2) )</td>
<td>( O(1) )</td>
<td>( O(\log_{2}(n)) )</td>
</tr>
</tbody>
</table>

The structure of the Transformer model is different from the traditional encoder-decoder structure. It does not use any RNN or CNN structure. It only contains the attention mechanism. The encoder is
composed of a stack of 6 identical layers. Each layer has two sub-layers. The first is a multi-head self-attention mechanism, and the second is a simple, position-wise fully connected feed-forward network. The decoder is also composed of a stack of 6 identical layers. In addition to the two sub-layers in each encoder layer, the decoder inserts a third sub-layer, which performs multi-head attention over the output of the encoder stack [1]. Comparing to RNN, the one which is used frequently in NMT, Self-attention can save a lot of computational complexity and reduce dependence distance, which increases the speed of parallel computing during training.

Every NMT model needs a lot of data to train before its translation result is acceptable. We used the English-Chinese parallel corpus including UN Conference, report on the work of the government of China, novels, new and so on which are provided by CWMT [7] and the Northeastern University or collected by ourselves from the Internet to train the model. The corpus size is approximately 7 Giga Bytes.

It is necessary to preprocess the corpus before training. First of all, we use regular matching to remove the serial numbers and labels that may be interference of training. Secondly, for Chinese corpus, segmentation is necessary. We use dynamic programming to find the maximum probability path, finding out the maximum segmentation based on word frequency from known words, and use the HMM-based open source word segmentation method to segment unknown words. All words are split by spaces.

After preprocess of the corpus, training can be started. We train the model on a server with a GPU (model NVIDIA Titan X). The training batch size is 1024 and the initial learning rate is 0.2. A checkpoint of training model is saved on every 1000 steps of training. Each output checkpoint model is about 1.2 Giga Bytes in size. In total, we trained it for 250,000 steps which took about 10 days. The loss during the training phase is shown in Figure 4.

![Figure 4. Loss chart of the training.](image)

After the training, the loss size was about 0.9220, and the evaluation loss of the test set was about 4.504. For general statements, the model can give translations that roughly match the meaning of the sentence. Moreover, it can also perform well for the unseen vocabulary. For example, we give out the name of the uncommon place name “Nancheng” (in English) in the test. Since the vocabulary does not include the name of this place, the model translate it to “Nanjing” (in Chinese). It can be seen that in the case where the dataset is not very large (the size of our corpus is far away from those for commercial usage), the model can still find the relevance between the words in vocabulary list and the unseen words.
Results and Usage Scenarios

We implement the local terminal on the hardware platform of Raspberry Pi 3 based on Android Things. The UI for real-time translation between Chinese and English is shown in Figure 5.

![Figure 5. User interface of the terminal.](image)

In the use scenario, the working modes of the hardware and software parts of the system are shown in Figure 6.

![Figure 6. Usage scenario graph.](image)

This system can display subtitle translation in real time. The output can be displayed externally. The results of recognition and translation are presented to the projection in the form of subtitles. The translation of the system supports Chinese and English. In the Chinese speech recognition part, the system supports Mandarin, Sichuan dialect and Cantonese dialect, which attributes to the Baidu ASR open platform.

This system can be used in lectures, conferences, classes, and other scenarios where consecutive interpretation is needed. Real-time bilingual subtitles can also be displayed on computer projections by desktop client. The desktop client of the system will also be used to interact with users. The client generates the QR code used for the conference participation and display it on the projection. Audiences can join the conference by scanning the QR code. The QR code takes users to a web client, which provides users the translation of his or her mother language and sentences spoken by speakers on their own devices. Users can also look over the records of sentences, and download formed documents according to their own requirements.

This system can be used as a kind of auxiliary equipment for interpreter. As the interpreters face the problems of fast speaking, limited bilingual processing ability and limited information capacity in the
brain, it will be good enough for them to translate 80% of the speech content (the interpretation with 90% to 100% accuracy is almost impossible for human interpreters in a short time). However, this system can help interpreters recall the interpretation results by viewing the translation of the system. Interpreters use the translated results of this system as a reminder for their own translation, or directly edit the output results from the system to make it the final translation result.

**Evaluation**

We evaluated transformer model of the system with BLEU. The BLEU is a method of automatic machine translation evaluation that is quick, inexpensive, and language-independent [8]. We also tested and evaluated the delay of the system’s translation effect.

For the BLEU evaluation, we got that the score of the model of translation of this system is 12.72. This is not really high due to the lack of corpus. For the delayed test, we used the report on work of government of China (2018) audio as voice input to compute the average translation delay of the system. The translation delay of the system refers to the time between the speaker completed a sentence and the result of the translation displayed. The value is about 0.8 second according to our test on 100Mbps network connection.

**Conclusion and Future Work**

In conclusion, this kind of system can be used in the consecutive speech scenarios such as lectures and conference where translation precious requirement is not high. It is faster than human interpreter. And in scenarios where the precious requirement is high, it can be the assistive device of human interpreter to give him or her a hint of what the speaker is speaking and the interpreter can directly edit the translation result provided by the system and use it as the final translate result.

In the future, we will try to put the translation result of system itself and the human interpreter edited result into the corpus of this system in order to enlarge it, which is known as increment learning. As corpus is updated and increasing constantly, the regular training of the NMT model of the system will be done to improve the quality of translation. We will also add TTS to the system in the future to make translated texts directly be spoke out in the language selected by the listener.

For extra function, we are also going to improve this system in another way. This system can not only be used for translation but also be an assistive system for the people with hearing loss. These people cannot join in lectures without sign language translator. But through this system, the subtitle of the speaker’s speaking will be directly displayed on the projection. This is also a great help for the people with hearing loss.

**References**


