IJIRCCE

e-ISSN: 2320-9801 | p-ISSN: 2320-9798

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

Volume 9, Issue 7, July 2021

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 7.542

International Journal of Innovative Research in Computer and Communication Engineering

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 7.542 |

|| Volume 9, Issue 7, July 2021 ||

| DOI: 10.15680/IJIRCCE.2021.0907150 |

Chat Application with Translation using NLP

Dhinesh GV, Sandeep S, Lokesh Kumar S

Department of Computer Science & Engineering, Velammal Engineering College, Chennai, India

Department of Computer Science & Engineering, Velammal Engineering College, Chennai,India

Department of Computer Science & Engineering, Velammal Engineering College, Chennai,India

ABSTRACT: These days, social media is very popular, especially chatting, people used to chat with each other through mobile phones rather than voice calls. According to textmagic: us smartphone users send and receive five times more texts than they make and receive calls. 3 out of 10 consumers would give up phone calls to use messaging. Over 68% of consumers said they text more than they talk on their smartphones. Used messaging apps enable relatively faster communication than email and have adapted to people's lifestyles. Apps such as facebook messenger and whatsapp are popular worldwidely. Their monthly number of active users is more than 1.3 billion and people recognize the messenger apps as indispensable tools for communication. Such messenger apps can be roughly classified into two types: personal chat and group chat. Regarding the personal chat, individuals can exchange one-on-one conversations and share private content and sensitive information. On the other hand, as to the group chat, users can share information with a large number of people instantly, and it is very handy for from small to large groups. Almost all of the messengers consist of these two kinds of systems. At present, various apps derive from these systems, and slack, upon which users can communicate with each other by multiple channels dependingonthetopics,hasemerged.Though,there are certain limitations in chatting. Two persons can chat if and only if both of their languages are common. But what if we can speak with a person who doesn't know our language, it will beawesome right. Such an application connects people who used to speak different languages without any diversities. Our proposed solution is to make a seamless end to end chatting application with end-to-end translation with privacy as the basic requirement.

I.INTRODUCTION

On the Internet, chatting is talking to other people who are using the Internet at the same time you are. Chat Applications are primarily meant for chatting with persons either one-on-one or in a group.Mostly these chat applications support most common languages like English, French, Chinese etc. But if a person wants to chat in their regional languages they would like to translate the received message through a translator and reply to those messages with the use of a translator because language is a big barrier when it comes to chat applications particularly for people who know only their regional languages. This problem has been solved in this chat application. This application has a translation feature which translates the chat messages being sent to other people either in-person or in a group chat.

The messages sent here are end-to-end encrypted, so privacy is maintained.

II.MACHINE TRANSLATION

Machine Translation (MT) is a subfield of computational linguistics that is focused on translating text from one language to another. With the power of deep learning, Neural Machine

Translation (NMT) has arisen as the most powerful algorithm to perform this task. While Google Translate is the leading industry example of NMT, tech companies all over the globe are going all in on NMT.This state-of-the-art algorithm is an application of deep learning in which massive datasets of translated sentences are used to train a model capable of translating between any two languages. With the vast amount of research in recent years, there are several variations of NMT currently being investigated and deployed in the industry. One of the older and more established versions of NMT is the Encoder Decoder structure. This architecture is composed of two recurrent neural networks (RNNs) used together in tandem to create a translation model. And when coupled with the power of attention mechanisms, this architecture can achieve impressive results.

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 7.542 |

|| Volume 9, Issue 7, July 2021 ||

| DOI: 10.15680/IJIRCCE.2021.0907150 |

III.TRANSLATION MODEL

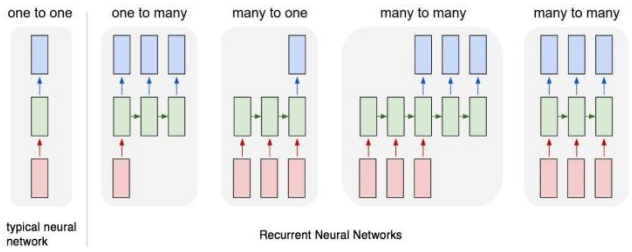
To translate a corpus of English text to French or to any other languages, we need to build a recurrent neural network (RNN). RNNs are designed to take sequences of text as inputs or return sequences of text as outputs, or both. They're called recurrent because the network's hidden layers have a loop in which the output and cell state from each time step become inputs at the next time step. This recurrence serves as a form of memory. It allows contextual information to flow through the network so that relevant outputs from previous time steps can be applied to network operations at the current time step.

This is analogous to how we read. As you read this post, you're storing important pieces of information from previous words and sentences and using it as context to understand each new word and sentence.

Other types of neural networks can't do this (yet). Imagine you're using a convolutional neural network (CNN) to perform object detection in a movie. Currently, there's no way for information fromobjectsdetectedinpreviousscenestoinform the model's detection of objects in the current scene. For example, if a courtroom and judge were detected in a previous scene, that information could help correctly classify the judge's gavel in the current scene, instead of misclassifying it as a hammer or mallet. But CNNs don't allow this type of time-series context to flow through the network like RNNsdo.

IV.RNN SETUP

Depending on the use-case, you'll want to set up your RNN to handle inputs and outputs differently. For this project, we'll use a many-to-many process where the input is a sequence of English words and the output is a sequence of French words (fourth from the left in the diagram below).



Each rectangle is a vector and arrows represent functions (e.g. matrix multiply). Input vectors are in red, output vectors are in blue and green vectors hold the RNN's state (more on thissoon).

From left to right: (1) Vanilla mode of processing without RNN, from fixed-sized input to fixed-sized output (e.g. image classification). (2) Sequence output (e.g. image captioning takes an image and outputs a sentence of words). (3) Sequence input (e.g. sentiment analysis where a given sentence is classified as expressing positive or negative sentiment). (4) Sequence input and sequence output (e.g. Machine Translation: an RNN reads a sentence in English and then outputs a sentence in French).(5) Synced sequence input and output (e.g. video classification where we wish to label each frame of the video). Notice that in every case there are no pre-specified constraints on the lengths sequences because the recurrent transformation (green) is fixed and can be applied as many times as we like.

V.PRE-PROCESSING

Here is a sample of the data. The inputs are sentences in English; the outputs are the corresponding translations in French.

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 7.542 |

|| Volume 9, Issue 7, July 2021 ||

| DOI: 10.15680/IJIRCCE.2021.0907150 |

```
English sample 1:  new jersey is sometimes quiet during autumn , and it is snowy in april .
French sample 1:  new jersey est parfois calme pendant l' automne , et il est neigeux en avril .

English sample 2:  the united states is usually chilly during july , and it is usually freezing in november .
French sample 2:  les états-unis est généralement froid en juillet , et il gèle habituellement en novembre .

English sample 3:  california is usually quiet during march , and it is usually hot in june .
French sample 3:  california est généralement calme en mars , et il est généralement chaud en juin .

English sample 4:  the united states is sometimes mild during june , and it is cold in september .
French sample 4:  les états-unis est parfois légère en juin , et il fait froid en septembre .

English sample 5:  your least liked fruit is the grape , but my least liked is the apple .
French sample 5:  votre moins aimé fruit est le raisin , mais mon moins aimé est la pomme .
```

When we run a word count, we can see that the vocabulary for the dataset is quite small. This was by design for this project. This allows us to train the models in a reasonable time.Next, we need to tokenize the data — i.e., convert the text to numerical values. This allows the neural network to perform operations on the input data. For this project, each word and punctuation mark will be given a unique ID. (For other NLP projects, it might make sense to assign each character a unique ID.).Whenwe run the tokenizer, it creates a word index,whichisthenusedtoconverteachsentenceto avector.

When we feed our sequences of word IDs into the model, each sequence needs to be the same length. To achieve this, padding is added to any sequence that is shorter than the max length (i.e. shorter than the longest sentence).One-Hot Encoding (not used)

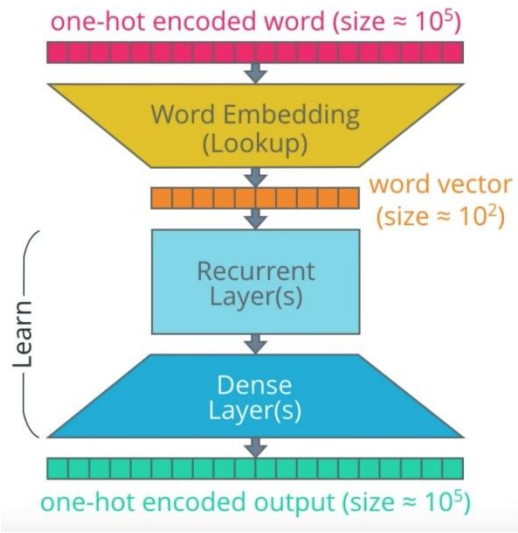
```
Sequence 2 in x
  Input:  [10 11 12  2 13 14 15 16  3 17]
  Output: [10 11 12  2 13 14 15 16  3 17] no padding
Sequence 3 in x
  Input:  [18 19  3 20 21]
  Output: [18 19  3 20 21  0  0  0  0  0]  padding
```

In this project, our input sequences will be a vector containing a series of integers. Each integer represents an English word (as seen above). However, in other projects, sometimes an additional step is performed to convert each integer into a one-hot encoded vector. We don't use one-hot encoding (OHE) in this project, but you'll see references to it in certain diagrams (like the one below).

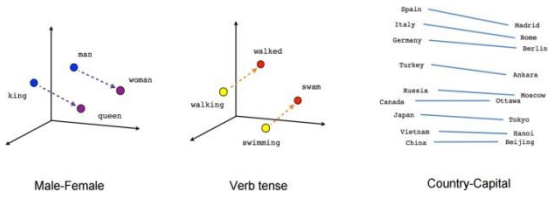
| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 7.542 |

|| Volume 9, Issue 7, July 2021 ||

| DOI: 10.15680/IJIRCCE.2021.0907150 |



Embeddings allow us to capture more precise syntactic and semantic word relationships. This is achieved by projecting each word into n-dimensional space. Words with similar meanings occupy similar regions of this space; the closer two words are, the more similar they are. And often the vectors between words represent useful relationships, such as gender, verb tense, or even geopolitical relationships.



Training embeddings on a large dataset from scratch requires a huge amount of data and computation. So, instead of doing it ourselves, we'd normally use a pre-trained embeddings package such as GloVe or word2vec. When used this way, embeddings are a form of transfer learning. However, since our dataset for this project has a small vocabulary and little syntactic variation, we'll use Keras to train the embeddings ourselves.

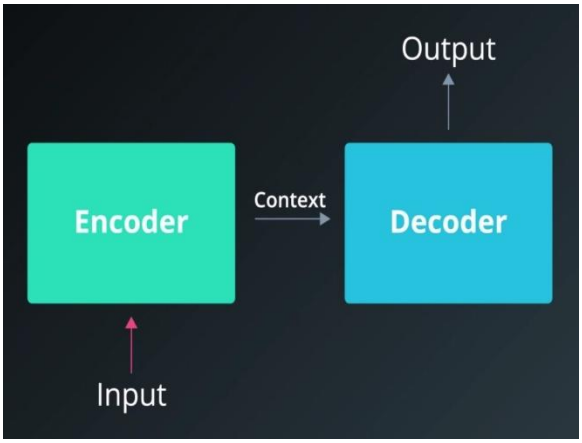
Encoder & Decoder

Our sequence-to-sequence model links two recurrent networks: an encoder and decoder. The encoder summarizes the input into a context variable, also called the state. This context is then decoded and the output sequence is generated.

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 7.542 |

|| Volume 9, Issue 7, July 2021 ||

| DOI: 10.15680/IJIRCCE.2021.0907150 |



Since both the encoder and decoder are recurrent, they have loops which process each part of the sequence at different time steps. To picture this, it's best to unroll the network so we can see what's happening at each time step.

In the example below, it takes four timesteps to encode the entire input sequence. At each time step, the encoder "reads" the input word and performs a transformation on its hidden state. Then it passes that hidden state to the next time step. Keep inmind that the hidden state represents the relevant context flowing through the network. The bigger the hidden state, the greater the learning capacity of the model, but also the greater the computation requirements. We'll talk more about the transformations within the hidden state when we cover gated recurrent units(GRU).

For now, notice that for each time step after the first word in the sequence there are two inputs: the hidden state and a word from the sequence. For the encoder, it's the next word in the input sequence. For the decoder, it's the previous word from the output sequence.

Also, remember that when we refer to a "word," we really mean the vector representation of the word which comes from the embedding layer.Here's another way to visualize the encoder and decoder, except with a Mandarin input sequence.

Bidirectional Layer

Now that we understand how context flows through the network via the hidden state, let's take it a step further by allowing that context to flow in both directions. This is what a bidirectional layer does.

In the example above, the encoder only has historical context. But, providing future context can result in better model performance. This may seem counterintuitive to the way humans process language since we only read in one direction.
However, humans often require future context to interpret what is being said. In other words, sometimes we don't understand a sentence until an important word or phrase is provided at the end.To implement this, we train two RNN layers simultaneously.

Hidden Layer with Gated Recurrent Unit (GRU).Nowlet's make our RNN a little bit smarter. Instead of allowing all of the information from the hidden state to flow through the network, what if we could be more selective? Perhaps some of the information is more relevant, while other information should be discarded. This is essentially what a gated recurrent unit (GRU)does.

VI.CONCLUSION

This Chat Application with translation explained in this report uses the efficient and accurate method in an open environment that offers people to chat with their regional language without any language barriers.

This web application is secure enough, reliable and available for anyone. It's end to end encrypted so your messages are

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 7.542 |

|| Volume 9, Issue 7, July 2021 ||

| DOI: 10.15680/IJIRCCE.2021.0907150 |

safe and no one can read your messages.

VII.FUTURE WORK

In future, the web applications' UI will be improved, some more extra features will be included and a mobile application on both Android and IOS will be launched.

REFERENCES

[1] Professional chat application based on natural language processing, S Karthick, R John Victor, S Manikandan, Bhargavi Goswami, 2nd February 2018Multi-User Chat Application. R. Gayathri,C.Kalieswari. June 2020

[2] Developing an End-to-End Secure Chat Application Noor Sabah,Jamal Mohammed,Ban N.Dhannoon November2017

[3] Secure Chat Application Based on Pure Peer-to-Peer Architecture. Mohammed Afendee, Abdhullah Mohammed, Mustafa Man. May2015

[4] UDP Based Chat Application. Akshit Malhotra, Vaibhav Sharma, Prateek Gandhi, Dr. Neetesh Purohit. 17 June2010

[5] Feasibility Study of Developing Chat Application in Bhutan. Sonam Deki, Kinzang Chedup, Tshering Choden, Sonam Wangda, Tandin Wangchuk. 29 November2018

[6] Development of Offline Chat Application: Framework for Resilient Disaster Management. Oliver M. Junio, Enrico P. Chavez. 15 April2019.

[7] Professional Chat Application based on Natural Language Processing. Karthick S, R John Victor, Manikandan S, Bhargavi Goswami. 04 June2018.

[8] Development and Evaluation of Mobile Application for Room Rental Information WithChat and Push Notification. Sonya R. Manalu, Aswin Wibisurya, Natalia Chandra, Alan Patrick Edijanto. 18 May2017.

[9] A Cryptographic Approach for Secure Client - Server Chat Application using PublicKey Infrastructure (PKI). Karabey, Gamze Akman. 16 February 2017.

[10] On Design and Implementation a Federated Chat Service Framework in Social Network Applications. Shi-Cho Cha, Zhuo-Xun Li,Chuan-Yen Fan, Mila Tsai, Je-Yu Li, Tzu-Chia Huang. 12 December2019.

[11] Mobile Chatting Server for GPRS networks. UsmanAli, TasnimAhmed. 12 May2008.

[12] A Chat Application in Lift. David Pollak, Steve Vinoski. 07 June2010.

[13] https://www.npmjs.com/package/socket.io/

[14] https://material-ui.com/

INNO SPACE
SJIF Scientific Journal Impact Factor
Impact Factor: 7.542

doi cross ref

ISSN INTERNATIONAL STANDARD SERIAL NUMBER INDIA

निस्केयर NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

9940 572 462 6381 907 438 ijircce@gmail.com

Scan to save the contact details