VISVESVARAYA TECHNOLOGICAL UNIVERSITY

BELAGAVI, KARNATAKA-590018

A Seminar Report on

"Understanding chat messages for stickers recommendation in hike messanger"

Submitted in partial fulfillment towards Seminar Work of VIII semester of

Bachelor of Engineering

in

Computer Science and Engineering

Submitted by

Prakruthi S **4GW16CS415**

Under the Guidance of

Asharani M

Assistant Professor

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Accredited to NBA, New Delhi, (Validity: 01.07.2017- 30.06.2020)

GSSS INSTITUTE OF ENGINEERING & TECHNOLOGY FOR WOMEN

(Affiliated to VTU, Belagavi, Approved by AICTE, New Delhi & Govt. of Karnataka)

K.R.S ROAD, METAGALLI, MYSURU-570016

2018-19

Geetha Shishu Shikshana Sangha (R)

GSSS INSTITUTE OF ENGINEERING & TECHNOLOGY FOR WOMEN

(Affiliated to VTU, Belagavi, Approved by AICTE, New Delhi & Govt. of Karnataka)
K.R.S Road, Mysuru, Karnataka-570016

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Accredited to NBA, New Delhi, (Validity: 01.07.2017- 30.06.2020)

2018-19

CERTIFICATE

Certified that the Seminar titled **"Understanding chat messages for sticker recommendation in hike messanger"** is a bonafide work carried out by **Prakruthi S(4GW16CS415)** in the partial fulfillment towards Seminar of VIII semester of **Bachelor of Engineering** in **Computer Science and Engineering** of the Visvesvaraya Technological University, Belagavi, during the year 2018-2019. The report has been approved as it satisfies the academic requirements.

Signature of the Guide	**Signature of the HOD**
(Mrs. AshaRani M)	**(Dr. S Meenakshi Sundaram)**
Assistant Professor	Professor and Head
Dept. of CSE	Dept. of CSE
GSSSIETW, Mysuru	GSSSIETW, Mysuru

ACKNOWLEDGMENT

I sincerely owe my gratitude to all the persons who helped and guided me in completing this Seminar.

I am thankful to **Mrs. Vanaja B Pandit**, *Honorary Secretary*, GSSSIETW, Mysuru, for having supported in my academic endeavours.

I am thankful to **Dr. Shivakumar M**, *Principal*, GSSSIETW, Mysuru, for all the support he has rendered.

I thank **Dr. Meenakshi Sundaram S**, *Professor and Head*, Department of Computer Science and Engineering, for his constant support and encouragement throughout the tenure of this Seminar work.

I would like to sincerely thank my guide **Mrs. AshaRani M**, *Assistant Professor*, and Department of Computer Science and Engineering, for providing relevant information, valuable guidance and encouragement to complete this Seminar work.

I am extremely pleased to thank my parents, family members and friends for their continuous support, inspiration and encouragement, for their helping hand and also last but not the least, I thank all the members who supported directly or indirectly in the academic process.

Prakruthi S

ABSTRACT

Stickers are popularly used in messaging apps such as Hike to visually express a nuanced range of thoughts and utterances and convey exaggerated emotions. However, discovering the right sticker at the right time in a chat from a large and ever expanding pool of stickers can be cumbersome. In this paper, we describe a system for recommending stickers as users chat based on what the user is typing and the conversational context. We decompose the sticker recommendation problem into two steps. First, we predict the next message that the user is likely to send in the chat. Second, we substitute the predicted message with an appropriate sticker. Majority of Hike's users transliterate messages from their native language to English. This leads to numerous orthographic variations of the same message and thus complicates message prediction. To address this issue, we cluster the messages that have the same meaning and predict the message cluster instead of the message. We propose a novel hybrid message prediction model, which can run with low latency on low end phones

TABLE OF CONTENTS

TABLE OF FIGURES

Chapter 1

INTRODUCTION

Hike Messaging app is the cross platform instant messaging app for communication on the internet. It is a free software available on various mobile platforms. It is one of the first instant messaging app made in India. Users can send each other graphical stickers, emotions, images, videos, audios, messages, contacts and user location.

Emoji's, gifs and stickers are extensively used to visually express thoughts and emotions. These go a long way to make chatting more productive and more fun. Hundreds of thousand stickers are available for free download. Once a user download sticker pack it gets added to a palette.

1.1 Desiderata for type-ahead Sticker Recommendation(SR)

The goal of type ahead sticker recommendation is to help user discover the perfect stickers. The latency of generating such sticker recommendation should be in tens and milliseconds. Hike users use low end mobile phones also such a system should start recommending new stickers as soon as they are added to the sticker store.

We need a solution which is efficient both in terms of CPU load and memory requirements. Since there are so many ways of expressing the same thing in chat, it's hard for any person to capture all variants of an utterance as tags.

1.2 Decomposing Sticker Recommendation

With the help of supervised model, which learns the most relevant stickers for a given context defined by the previous message. Massive skew in historical usage toward a handful of popular stickers, it becomes difficult to collect reliable, unbiased data to train such end-to-end model will need to be retained frequently to support stickers and updated model.

For message prediction we train a classification model with the help of historical chat data how efficiently set up message prediction task. Orthographic variations of chat messages analyses a framework for message clustering, which can identify different messages that have the same meaning. Choosing one of top k messages helps us drastically reduce the number of classes .Computational limitations on low end smart-phones has large fraction

of our users use Hike on low end mobile devices with severe limitations on memory and compute power.

1.3 Countless expression in chat

Our users prefer to use native a languages when it comes to chatting .Due to lack of effective local keyboard support, users tend to translate messages from their native languages using English keyboard. Here we use acronyms e.g. “where are you >>”whr r u”.repetition of characters to emphasize certain words. Message embedding clustering is different approaches to train embedding for chat messages and study their efficacy in learning similar dense representation for messages that have the same intent.

In Hybrid message prediction model only tric component has to be executed for each character, hence the system will be able to meet the latency constraint easily. Scores from these components were combined to produce final scores for message.

Novel based application of type-ahead sticker recommendation decompose the recommendation in two steps message prediction and sticker substituation.Cluster messages that have the same meaning create dense representations for chat messages, these can run with low latency on low end phones that have severe computational limitations.

Chapter 2

LITERATURE SURVEY

- Hike was launched on Dec 2012.
- Attained 15 Billion users in 2014.
- Over next few months it launched free text messaging across India .
- Made its first acquisition in 2015, buying free voice calling company Zip Phones which allowed it to launch free app based calling services.
- Integrated Tiny Mogul and Hopper under its own brand name.
- In 2015 it launched “Great Indian Sticker Challenge” to invite designers to design new stickers.
- In October 2015 it reached 20 Million users with 20 billion messages shared each month.
- In January 2016 it crossed 100 Million users
- In November 2016 it launched the feature of stories
- In June 2017, Hike launched Hike 5.0 along with Hike Wallet, first indian messaging app to launch this feature
- Later in 2017, Hike announced its acquisition hardware startup creo
- In January 2018, launched Total, Built by hike. It allowed users to access services on their mobile even in the offline mode.

Chapter 3

METHODOLGY

We recommend sticker task into two steps. First, we predict the message that user is likely to send based on chat context that user typed inputs. Second we recommend stickers by mapping the predicted message to stickers that can substitute it.

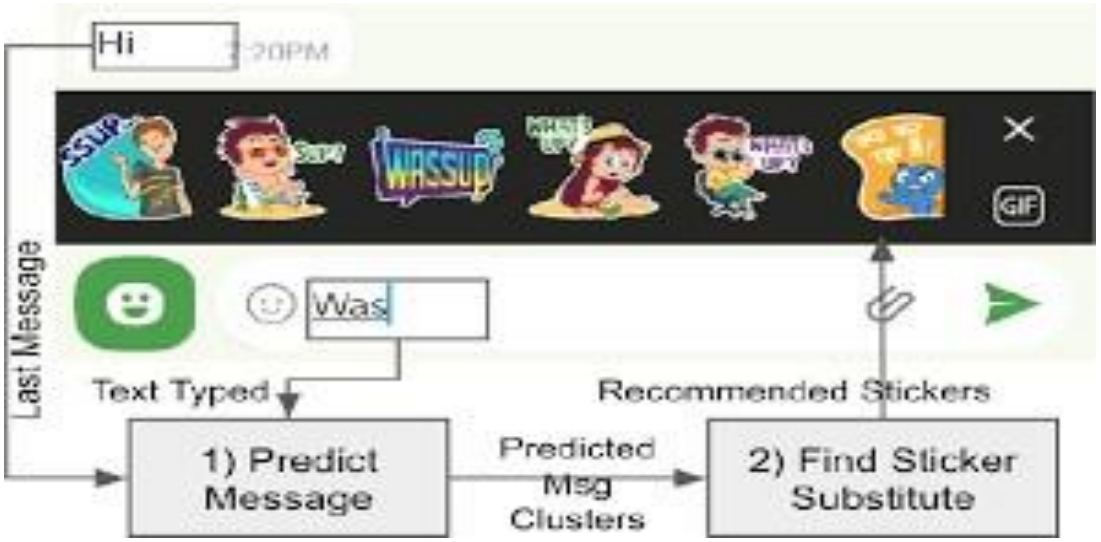

FIG 3. Sticker Recommendation UI on Hike and flow of two step sticker recommendation system.

3.1 Chat Message Clustering

Cluster top messages in our chat so as to use them as classes in the message prediction model. For covering large amount messages in our chat with a limited amount of clusters, we should be able to group all similar meaning messages into single cluster. This has to be done without compromising quality of individual clusters.

We describe a encoder which is used to encode chat phrases into dense vector representation .We explore two network architecture which are used to train the message embedding such that we learn similar representation for messages that have same meaning.

Encoder for chat message that effectively captures its semantics and represents it as fixed length vector. Input the message *mi* , consisting of sequence of N words and the output

is continuous vector $e_i \in {}^{dX1}_{R}$. We use a character CNN that leverage sub word information to

learn similar representations for orthographic variants of the same word. Let V_C be the vocabulary of all characters and d_c be the dimensions of character embedding. For word w_t we have a sequence of characters obtained by the character level embedding in matrix .

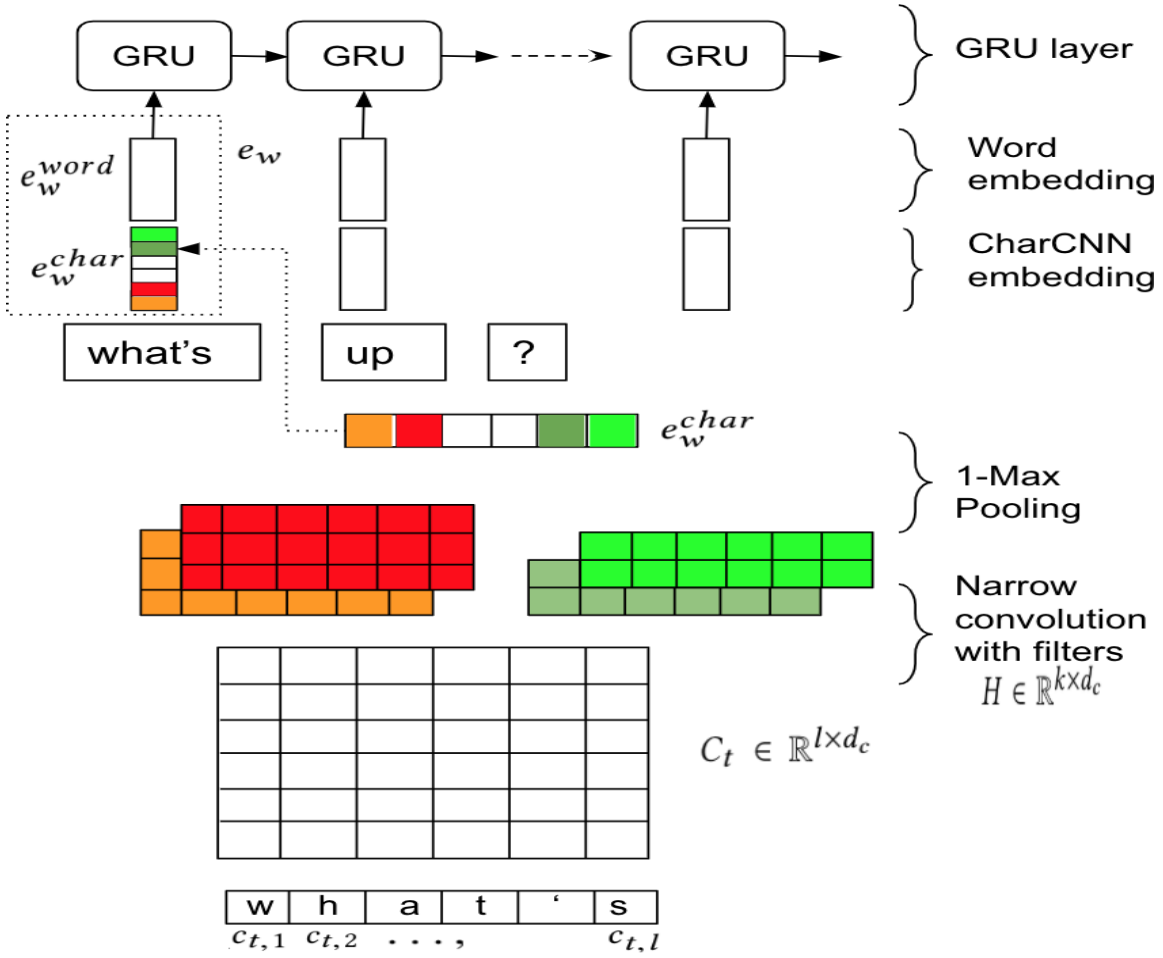

FIG 2.1.1 Illustration of encoder comprises of character based CNN layer for each word

3.2 Model Architecture

Two model architectures to train the encoder , one inspired from Skip-Thought where surroundings sentences are generated from the current sentence embedding, Second is a discriminative approach using a Input response model where network is optimized to score the gold reply message higher compare to other reply messages.

3.2.1 Skip Thought Chat (STC)

The input to our model is a triplet m_{i-1}, m_i, m_{i+1} extracted from a conversation between two users such that if m_i is a message sent by the user, then m_{i-1} and n are the messages by the user sent by that user before and after sending m_i respectively. Message mi gets encoded using encoder as described to get message embedding e_i.

Instead we use a feed forward layer to create a Bag of Words (BOW) representation for the surrounding i,e previous and next messages from the encoded message embedding e_i.

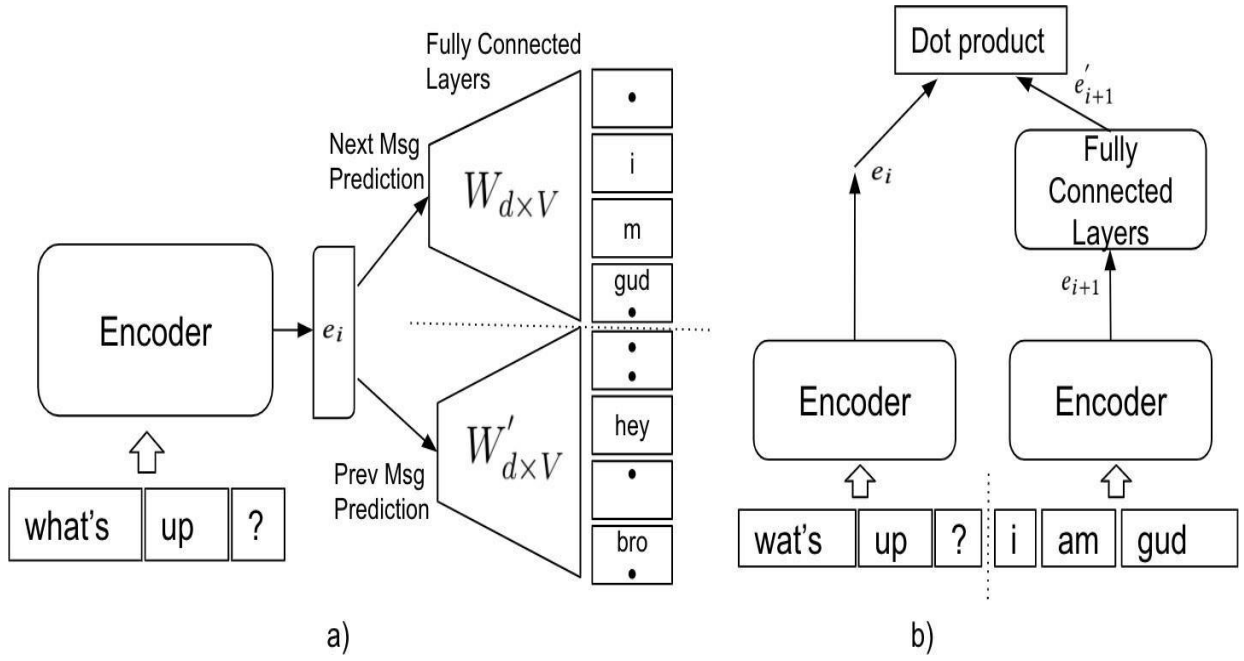

FIG 3.2.1 Model architecture for learning message embedding (a)Skip Thought Chat(STC) (b)Input-Response Chat

3.2.2 Input-Response Chat (IRC)

The input of our model is tuple m_i, m_{i+1}, extracted from a conversation between two users. Message m_i and m_{i+1} gets encoded using encoder. Both e_i and e_{+1} represents the encoding of message in the same space. To handle this uncertainty, we used HDBSCAN algorithm to cluster the phrases. The algorithm builds a hierarchy of clusters and handles the variable density of clusters in time efficient manner.

3.3 Message Cluster Prediction

Describing approach for predicting the message a user is going to send in reply of previous message. In chat, the next message to be sent it heavily influenced by conversation context. For now, we are using only last message from the other person as the context signal. When user starts typing, we should update our predictions accordingly as it gives a strong signal about the next message user going to send. Message prediction latency should be in order of few milliseconds so that user typing experience does not get affected due to recommendation processing time.

Neural network (NN) based classification model that accepts both signals is a good choice. But it is computationally expensive to do inference of neural network (NN) model on mobile devices which have low CPU memory. To overcome these problems, we evaluate two orthogonal approaches. One is to apply a quantization scheme to reduce the model size and second is to build a hybrid model, composed of a neural network component and a trie-based search.We have two approaches here First, Quantized Message Prediction Model and second, we have Hybrid Message Prediction Model

3.3.1 Quantized Message Prediction Model

Classification model for message prediction task where input is the previous message and typed text and output is the message cluster user is likely to send. Top G message clusters are considered as the classes for this model. One of the approach is to quantize the weights and activation of neural model from 32 bit float representation to lower bit representation. We used a quantization scheme detailed in that convert both weights and activation from 32-bit float number to 8-bit integer and few 32-bit bias integer. It reduces the model size by a factor of 4. When we quantize the weights of the network after training the model with full precision float numbers then performance during inference time reduced significantly.

We use a quantize aware training to reduce the effect of quantization on model accuracy. For that, during training too, we use the quantized weights and activation to compute the loss function of the network.

3.3.2 Hybrid Message Prediction Model

To reduce the size of model, we built a hybrid model in which a resource intensive component is separated out and its output is combined with a lighter on-device model. Figure 1.4 outlines

the flow of the hybrid model. The three different components are Reply model, Typed Model, Combiner model.

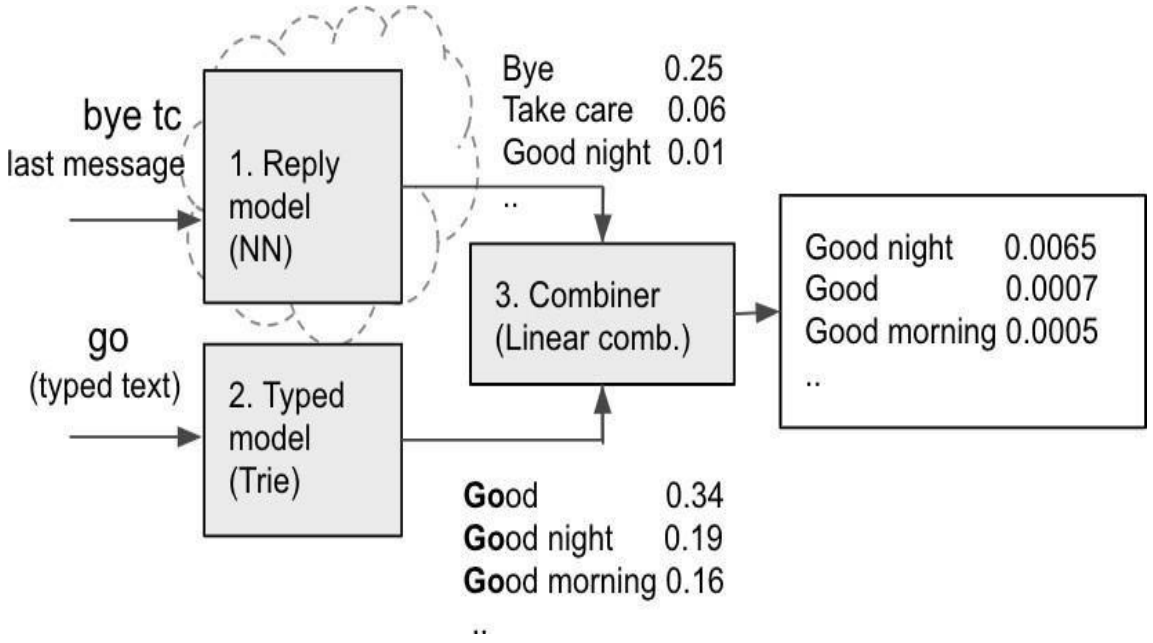

FIG 3.2.1 Hybrid architecture for message cluster prediction 1) Reply Model (NN) 2) Typed Model (Trie) 3) Combiner Model (Linear Combination)

3.3.1.1 Reply Model

Predicts the next message cluster using a neural network (NN) based classification model on server and sends the result to mobile device. We built a model whose input is last message only and output is message cluster the user is likely to send. The model produces probability score $P_{reply}(G=g|prev)$ where *g* denotes a message cluster from the set of message cluster G. Reply models queried only once for a message and results are sent to client delivering the message itself

3.3.1.2 Typed Model

For each character typed client predict the message cluster based on typed text using trie. Trie is an efficient data structure for retrieving values whose key starts with the given query string. By keeping <phrase,message cluster> pairs as <key,value> pairs in trie, we will be able to retrieve relevant message clusters by passing typed text as query to the trie. These <phrase,message cluster> pairs are nothing but chat messages and corresponding message clusters.

In order to reduce false positive results, we assigned a min prefix length for each phrase in the trie, so that a matching record is retrieved only if the typed text meets this min prefix length or its message cluster is present in recommendation available from the reply model. For example, Suppose phrase "hi" has a min prefix length set to 2, then a typed text "h" will not retrieve record corresponding to "hi", though it is matching to the typed prefix. But if message cluster of "hi" is present in the scores coming from reply model, then "hi" is retrieved even if length of the prefix is below the required minimum length 2.

3.3.1.3 Combiner Model

Combiner aggregates the scores from both models to obtain a final score for each message cluster. After receiving relevant message clusters and corresponding scores from reply model and trie model, a final score for a message cluster P_{hybrid} (G=g|typed,prev) is computed for a message weighted combination of P_{reply}(G=g|prev),P_{trie}(G=g|typed) and additional features such as string typed with appropriate transformations and including interactions terms between them.

3.3 Message cluster to Stickers Mapping

We make use of a message cluster to sticker mapping to suggest suitable stickers from the predicted message clusters. When a sticker is created, it is tagged with conversational phrases that the sticker can possibly substitute. We use this meta-data in order to map the message clusters to stickers. We computed similarity between the tag phrases of a sticker and the phrases present in each message cluster, after converting them into vectors using the encoder.

Compared to the historical approach of suggesting a sticker when the user's typed input matches one of its tag phrases, the current system is able to suggest stickers even if different variations of the tag phrase are typed by user, as we have many variations of a message already captured in the message clusters. We regularly refresh the message cluster to sticker mapping by taking into account the relevance feedback observed on our recommendations.

3.4 Message Prediction Evaluation

Our aim is to show the right stickers with minimal effort from user. So, we compared our hybrid model and quantized models mainly on the following metrics.

- Number of character that a user need to type for seeing correct message cluster in top 3 positions of Character to be typed.

- How many times the model has shown wrong message before predicting the correct message cluster in first3 positions

For training the message prediction using NN model, we collected pairs of current and next messages from complete conversation data. One is the prediction of next message based only on current message. For that we trained the model directly from current and next message pair instances where next message was mapped to its corresponding message cluster.

Second we build a trie model based only on typed message. The hybrid model was performing slightly better in terms of times inaccurate recommendations shown and fraction of messages retrieved. Compared to quantized models, hybrid model needed slightly more than one more character to be typed on an average get required predictions.

Method	# of Character to be Typed	# of times inaccurate predictions shown	Fraction of msg retrieved
Quantized NN (100d)	**1.58**	1.47	0.978
Hybrid	2.84	**1.22**	**0.991**

Table 2: Performance of different message prediction models

FIG 3.4 Performance of different messages prediction models

Chapter 4

APPLICATIONS

Stickers are the one of the easiest way for conversation, it gives better and unique way for chatting. It is one of the best method of expressing the emotions, some of the applications are as follows

- One of the best and effective way of communication.
- It also can be used where language is barrier.
- Good for messaging applications where expressions plays a vital role.
- Unique chat themes, sticky for other messaging apps i.e. we can use and share the stickers of hike in other messaging apps such as whatsapp, line wechat etc.
- Consists a lot of interesting and useful stickers along with emoji's of almost every type.
- This is the only messenger which has regional sticker which is not available in any other messenger.
- It is also introduce latest sticker time to time on occasion like festival, sports, etc

Chapter 5

ADVANTAGES

- Personalize your message, increases the message clarity and even conveys the emotions
- Sticker Saves Space & Time.
- Optimal use of Stickers in messages leave a better impression in the minds of the users and it apparently boosts your user engagement.
- Stickers are more appropriate than text.
- Abundant varieties of Stickers for all emotions.
- Appropriate recommendations or suggestions during conversation.
- Easy to download and use.

5.1 DISADVANTAGES

- Sticker sharing option was removed which made people to use stickers from in other apps.
- Sometimes sticker suggestions will be inappropriate with the context.
- It consumes lot of space.
- Downloading of more stickers consumes more memory on a low end phones.

CONCLUSION

The main aim of this technical seminar is to understand a novel system for deriving contextual type–ahead sticker recommendation within a chat application. We decompose the recommendation task into two steps. First we predict the next message that a user is likely to send based on the last message received in the chat. As the user types the message, we continuously update our predication by taking into account. Second, we substitute the predicted message with the relevant stickers. We describe a clustering solution that can identify, message clustering reduces the complexity of the classifier used in message prediction.

For message prediction on low-end mobiles phones, we use hybrid model that combines a neural network on the server and memory effective sticker recommendation .It helps in better discovery for message cluster which are present rarely in conversation data.

REFERENCES

[1] 2018. Multimodal emoji prediction. arXiv preprint arXiv:1803.02392 (2018).

[2] arXiv preprint arXiv:1810.04805 (2018).

[3] Study on a twitter based corpus. In Proceedings of the 8th Workshop.

[4] Computational Approaches to Subjectivity, Sentiment and Social Media Analysis.

[5] https://www.hike.in

[6] https://www.google.com