INTERNATIONAL JOURNAL FOR INNOVATIVE RESEARCH IN MULTIDISCIPLINARY FIELD
ISSN(O): 2455-0620 [Impact Factor: 7.581]
Monthly, Peer-Reviewed, Refereed, Indexed Journal with IC Value : 86.87
Volume - 9, Issue - 10, October - 2023 Publication Date: 25/10/2023

DOIs:10.2015/IJIRMF/202310018 --:-- Research Paper / Article / Review

Enhancing Short Text Conversations with an External Semantic Memory

Venkata Ramana Kaneti
Assistant Professor
VNR Vignana Jyothi Institute of Engineering and Technology, Hyderabad, Telangana, India – 500090.
kvramana_2001@yahoo.com

***Abstract*:** *Short Text Conversation (STC) is a simplified conversational task consisting of a one-round conversation with two short text sequences. The former sequence, typically provided by a human, is known as the "post," while the latter, generated by a computer, is referred to as the "comment." STC plays a crucial role in the development of open-domain conversation systems. The existing system relies on the traditional sequence-to-sequence approach for STC, which has its limitations. It often results in responses lacking diversity and substantial content. Moreover, controlling the topic or semantic relevance of the responses can be challenging when using this approach. Additionally, maintaining and managing the dataset can be time-consuming, and it consumes substantial storage space. Our proposed system introduces an innovative approach by incorporating an external memory to represent interoperable topics and semantics. In the realm of STC, there are two major frameworks: Retrieval-based methods and Generation-based methods. Generation-based methods involve training a text generation model on the STC data and generating responses based on the model's understanding of the given post. The core of our system utilizes a Sequence-to-Sequence model for language generation. This model consists of two key components: an encoder and a decoder. Specifically, we employ Gated Recurrent Units (GRU) for encoding and Recurrent Neural Networks (RNN) for decoding. Our approach called the "External Semantic Memory Guided Sequence-to-Sequence Generation Algorithm," is used to train the encoder-decoder model to maximize the likelihood of generating coherent and contextually relevant sequences. One of the notable advantages of our proposed system is its ease of dataset management and control. This approach significantly reduces the time and storage requirements for generating semantically meaningful responses in STC.*

Keywords: *Short Text Conversation, Sequence-to-Sequence, External Semantic Memory, Gated Recurrent Units, Recurrent Neural Networks, Language Generation, Dataset Management.*

1. INTRODUCTION

1.1 Introduction to Short Text Conversation

In recent years, the widespread use of social media, like Twitter and micro-blogs, has enabled data-driven approaches to Short Text Conversation (STC) [1]. STC involves short exchanges, often between humans (referred to as "post") and computers (referred to as "comment"). It contributes significantly to the field of STC. Two main approaches for STC are retrieval-based methods and generation-based methods [2][3][4]. Retrieval-based methods select relevant responses, while generation-based methods train models to generate comments, offering novelty compared to retrieval-based ones. Retrieval-based STC is similar to Information Retrieval (IR) tasks, whereas generation-based methods offer coherent but often generic responses due to a lack of grounding knowledge. The Recurrent Neural Network (RNN) sequence-to-sequence model is foundational for STC. It tends to produce uniform, less informative replies [5][6]. To address this, external semantic memory is introduced, enhancing diversity and manual semantic manipulation [10][11].

1.2 External Memory Guided Sequence to Sequence Learning:

External memory significantly enhances neural network performance across tasks like question answering, sorting, machine translation, and reasoning. It's akin to "data" in a computer's architecture, while "instructions" are typically in the model's weights. This approach suggests using memory to store specific instructions, like transformation rules in sequence-to-sequence learning, in an external memory accessible to both the neural network and human experts. This creates a novel learning paradigm where both concepts and rules are taught to the neural network.

1.3 Problem Definition

Short Text Conversation (STC) has gained prominence in recent years, especially on platforms like Twitter and microblogs, for

communication purposes involving posts and responses. The challenge of conveying meaningful messages within the constraint of a maximum of 140 characters is noteworthy. However, the existing systems primarily rely on the traditional sequence-to-sequence approach, which often falls short in terms of diversity and generating meaningful responses. This approach struggles to control the topics and the semantic quality of responses among multiple generated candidates. To address these issues, a novel generation-based approach is employed. This approach leverages RNN-based models to adaptively encode the input post into a fixed-length vector and subsequently employs a decoder to generate comments word-by-word. Generation methods involve training a text generation model on STC data, allowing the generation of fresh comments from the provided post, ultimately enhancing both diversity and substantive responses.

1.4 Existing System

Conventional Sequence-to-Sequence (Seq2Seq) models have been widely employed in the field of answer generation for conversations. However, the specific requirements for various communication contexts vary significantly. For instance, customer services demand precise and dependable responses, while chatbots favor a variety of responses to engage a diverse user base. The current Seq2Seq model often falls short of meeting these intricate requirements, as it relies on optimizing responses based on a general, average probability. Consequently, it frequently yields generic responses like 'I don't know.' This existing system utilizes a traditional sequence-to-sequence approach for Short Text Conversation (STC), resulting in limited diversity and a lack of substantial responses. This approach struggles with controlling the topic and semantic quality of responses among multiple generated candidates. Additionally, the dataset management presents challenges, making updates and control cumbersome, and the response generation process time-consuming. The disadvantages of the current system can be summarized as follows:

1. Repetition of user questions.
2. Difficulty in managing the dataset.
3. Lengthy and cumbersome dataset training and updates.
4. Time-consuming response generation.

1.5 Objectives

The main objective is to

1. Provide semantic reply for the post.
2. To achieve Diversity and Substantiality.
3. To optimize time and space for the post.
4. To improve the overall performance of the system.

1.6 Proposed system

The proposed framework introduces an external memory system to represent versatile topics and semantics in short text conversations. Short text conversation primarily employs two main frameworks: retrieval-based methods and generation-based methods. Retrieval-based methods seek existing statements from the STC training corpus, while generation methods train the STC model to generate new comments. The sequence-to-sequence model, comprising two encoder and decoder sections, is used for text generation. The encoder utilizes Gated Recurrent Unit (GRU), and the decoder employs Recurrent Neural Network (RNN). External Semantic Memory Guided sequence-to-sequence generation algorithm enhances decoder training to improve sequence generation likelihood. The encoder part encodes variable-length input into a fixed-length vector, and the decoder generates variable-length output word by word. However, this approach faces the vanishing gradient problem with long inputs. To address this, a soft setup mechanism enables the decoder to access all encoder's hidden vectors, ensuring relevant input components are considered during word generation. This system leverages an external semantic memory, which can be manually manipulated to introduce new semantics, offering rich diversity in comment generation. The thesis introduces a modern sequence-to-sequence approach to STC learning. The external semantic memory consists of a tensor comprising matrices that describe comment sentence semantics. Each matrix represents possible comment sentences related to a specific semantic key. These matrices form a basis for sentence embeddings, creating an extensive semantic comment space. During generation, a semantic key is extracted from the input sequence, used to create a comment sentence from the external memory, and combined with post sequence embedding in a sequence-to-sequence model to generate the final comment. This approach provides interpretative control over statement topics and semantics by manipulating the semantic keys.

The following are the advantages of the proposed system:

1. Repeated response can be avoided by the admin.
2. Semantic reply can be generated by the system.
3. Time consumption is reduced.

2. LITERATURE SURVEY

External memory systems have made a resurgence in deep learning [1][2], playing a significant role in the success of neural network-based systems. These memories extend the conventional representation of data instances, offering greater flexibility in instance representation format, space limitations, and information access methods. Unlike traditional computers, neural network instructions are developed through a learning process, with weights adjusted using learning algorithms. The "Guided Sequence-to-Sequence (GUIDED-SEQ2SEQ) learning" framework leverages memory to store parts of the instructions, particularly transformation rules. Additionally, it employs a hybrid addressing strategy for rule execution.

Creating computer systems for general-purpose conversations is a challenging task. While sequence-to-sequence (Seq2Seq) models have gained popularity in data-driven applications, they often produce short, generic, or incoherent responses. The Neural Responding Machine (NRM) is an answer generator based on neural networks that uses an encoder-decoder framework. NRM is trained on substantial one-round conversation data, producing grammatically correct and content-specific responses, outperforming other models in the same environment.

In natural language processing, neural network models have shown promise, yet their applicability in various tasks remains an ongoing challenge. Recent work has introduced new neural network models for tasks like machine translation, improving performance over traditional statistical machine translation (SMT) methods. These models, such as RNN Encoder-Decoder, achieve high-quality translations and sensible representations of phrases and sentences. Neural networks are expanding the boundaries of their use in text modeling.

Text generation is a key component in various natural language processing tasks, including machine translation, speech recognition, and image captioning. Long Short-Term Memory (LSTM) networks, a special type of Recurrent Neural Network (RNN), are particularly well-suited for this task, predicting the next term in a sequence. LSTM demonstrates potential in predicting the next point in a sequence. A proposed Bangla Text Generator with LSTM achieves satisfactory precision.

Communication between humans and computers is a challenging area of artificial intelligence, particularly in short text conversation. Retrieval-based and generation-based systems are two main approaches, but they have their limitations. An ensemble of open-domain retrieval and generation-based dialog systems has shown significant improvements, outperforming individual components.

In the field of natural language generation, Sequence-to-Sequence (Seq2Seq) models have been widely adopted. Language modeling, paraphrase detection, and word embedding are areas where deep neural networks have shown great success. However, generative models often face limitations, such as in handling long sentences. Innovative techniques like dilated CNNs and the use of external facts aim to overcome these challenges and have shown improvements in language modeling.

In short text classification, which has gained prominence due to the exponential growth of short text on the internet, classifying short text is challenging due to its sparse nature, syntactic structure, and colloquial language. There is a need for effective algorithms to address these challenges, but the existing algorithms lack comprehensive reviews.

Neural machine translation techniques have been introduced to create a single neural network that jointly optimizes translation outputs. These encoder-decoder models have shown great promise in improving the quality of translations. The inclusion of soft alignment mechanisms further improves translation quality.

The study of short text conversation focuses on developing models that can return rational responses to human messages. It leverages vast social media conversation data and introduces a retrieval-based model to perform this task effectively.

Task-oriented dialog systems have become more prevalent in recent years, and end-to-end models using RNNs are gaining popularity. The Hybrid Code Network (HCN) is an efficient end-to-end model, exemplified in a multi-language hospital receptionist robot.

The field of text generation has been advanced through the use of probabilistic language models and Sequence-to-Sequence (Seq2Seq) models. Seq2Seq models condition responses on the conversation history and external "facts," enhancing their versatility and applicability in open-domain settings.

Neural networks have demonstrated their effectiveness in tasks like object recognition and sentiment analysis. These networks can be applied to natural language processing tasks, with the proposed methods offering novel techniques to improve performance.

The Multi-Resolution Recurrent Neural Network, which models natural language generation as two parallel stochastic processes, has shown promise in improving the quality of responses in dialog systems. It demonstrates superiority in both the Ubuntu technical support domain and Twitter conversations.

Humans use complex memory systems to access and interpret important information. This concept has influenced the design of memory-augmented neural networks, such as LSTMs, which enhance memory capacity, deal with fading gradients, and allow the comparison of events over time.

3. SHORT TEXT CONVERSATION

3.1 Short Text Conversation (STC)

Natural language communication is one of the most difficult artificial intelligence issues, which includes language comprehension, reasoning, and the use of common-sense knowledge. Recently due to the stormy growth of microblogging services such as Twitter and Weibo, the amount of conversation data available on the web has extremely increased. That makes it possible to attack the conversation using a data-driven approach. Instead of multiple rounds of conversation, the task at hand, called Short-Text Conversation (STC), considers only one round of conversation in which two short texts form each round, the former being an input (referred to as a post) from a user and the latter a answer from the machine. The STC research could shed light on understanding the complicated mechanism of conversation in natural language. The short text cap is 140 characters; themeaning of the text is difficult to understand.

3.2 Sequence-to-Sequence Learning for STC

In the field of machine translation, sequence to sequence learning is also referred to as the encoder- decoder framework. The method employs a Recurrent Neural Network (RNN), usually a Long Short Term Memory Network (LSTM) or Gated Recurrent Unit (GRU) to map the input sequence to a fixed dimensionality vector, and then another RNN to decode the target sequence from the vector.

$$z_t = \sigma\,(w_z x_t + U_z h_{t-1}) \quad \text{---------- (1)}$$

$$\tilde{} h_t = \tanh\,(\mathrm{w}x_t + \mathrm{u}\,(r_t * h_{t-1}) \quad \text{----------} \qquad (2)$$

$$h_t = z_t * h_{t-1} + (1 - z_t) * h_t \qquad \text{----------(3)}$$

where x_t is the input word embedding vector input word at time t, ht is the hidden state vector output hidden state vector, $h_t$is the reset vector output, r_t is the update gate output vector, W and U are weight matrices and denotes element-wise product output. GRU will take a variable-length sequenceof vectors $\{x_1 \cdots x_T\}$ from the above formula, and convert them to a fixed-length vectorh_T. This recurring characteristic of the calculation makes it suitable for modeling sequences.

The encoder part in sequence-to-sequence training is usually a standard GRU for mapping a sequence into a context vector of a fixed-size sentence level. Conversely, the decoder takes the context vector as an additional input and generates word by word output sequence. With the exception of using the background vector is additional data, the decoder GRU formula is identical to equations (1) to (3). For example provided the encoder output vector h_T as the decoder equation (1) may be rewritten as

$$r_t = \sigma(w_r x_t + U_r h_{t-1} + C_r c_t) \text{ ------- (4)}$$

Where C is weight matrix of context vector and $c_t = h_t$.

Figure 1 explains the sequence to sequence approach in which it reads input from user and encodes theinput into fixed length vector, and generates sequence output in the form of normal text. By using generation based approach, it generates new reply for post.

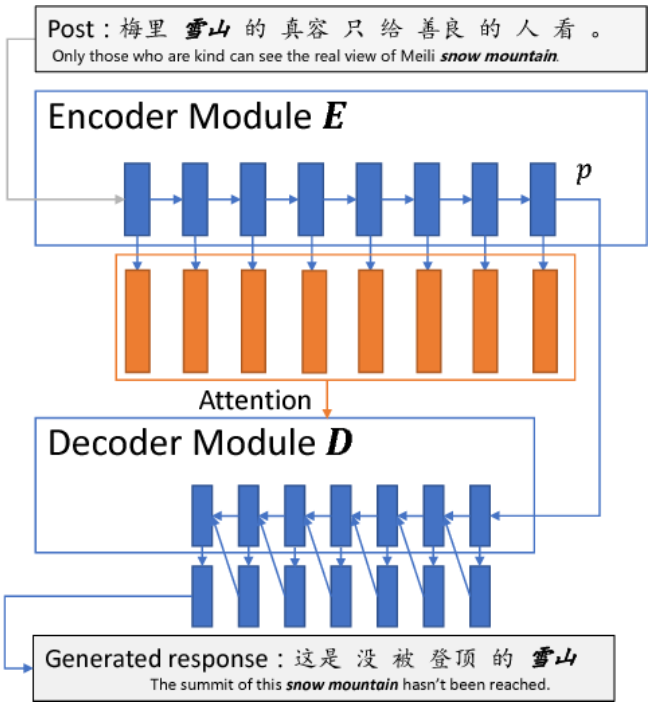


Figure 1. Sequence-to-Sequence Learning for Generation-Based STC

3.3 Input and Output Sequences have the Same Length

When both input sequences and output sequences have the same length, it can implement such models simply with LSTM or GRU layer. The example script shows how to teach a RNN learn to add numbers, encoded as character strings:

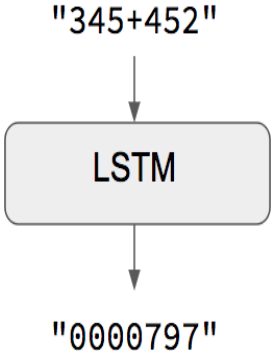


Figure 2. Example for Input and Output having Same Length

One caveat for this approach is that it assumes that target [... t] given input [... t] can be generated. That works in some cases (e.g. adding digit strings), but doesn't work in most cases of use. In the general case, information is needed on the entire input sequence to start generating the target sequence.

3.4 Input Sequence and Output Sequence having Different Lengths

In the general case, input sequences and output sequences have different lengths (e.g. machine translation) and the entire input sequence is required in order to start predicting the target. This requires a more advanced setup, which people commonly refer to when mentioning "sequence to sequence models" with no further context.

A layer of RNN serves as a "encoder" that processes the input sequence and returns its own internal state. Dispose of the RNN encoder outputs, only to recover the state. In the next step this state will serve as the decoder's "sense," or "conditioning."

Another RNN layer acts as a "decoder" it is trained to predict the next target sequence characters, given the previous target sequence characters. Specifically, it is learned to transform the target sequences into the same sequences but offset in the future by a one-time step, in this sense is a training method called "teacher pushing." Importantly, the encoder uses the state vectors from the encoder as the initial state, which is how the decoder gets details on what to produce.

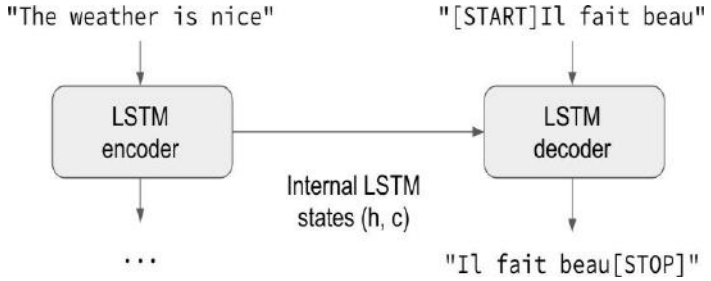


Figure 3. Example for Input and Output having Different Lengths

Figure 3 explains the example for input and output having different lengths, in which the length of the input is different from output. Here the input is “the weather is nice” converts it into standard encode form by using LSTM encoder and perform action on input and decode the standard input intonormal decoder form which is easily understandable by users. The LSTM decoder is used to predict the semantic reply.

3.5 Decode Unknown Input Sequences

It has some slight different process general process in which having same length and differentlengths

1. Encode the input sequence into state vectors.
2. Start with a target sequence of size 1 (just the start-of-sequence character).
3. Feed the decoder with the state vectors and 1-char target sequence to produce predictions aboutthe next character.
4. Sample the next character using these predictions (simply use argmax).
5. Append the sampled character to the target sequence6. Repeat until we generate the end-of-sequence character or we hit the character limit.

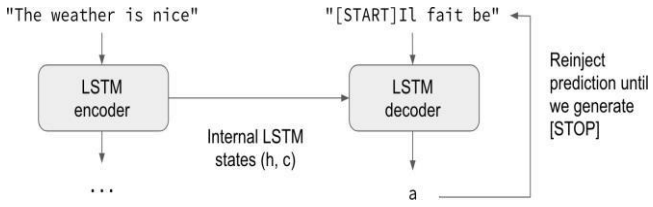


Figure 4. Example for Decode Unknown Input Sequence

Figure 4 illustrates an example of unknown input sequence decoding in which the user receives theinput "the weather is nice." Using LSTM encoder, converts the input into standard encoded form. Repeat the process until the exact decoder for an encoder is predicted. Using LSTM decoder generates correct semantic response.

3.6 External Memory Guided Sequence to Sequence Learning:

A tensor is constructed in the form of a list of matrices to represent the semantics of the comment sentences, referred to as external semantic memory. Each matrix represents all possible sentences of comments that correspond to a specific semantic key. Each of the matrix's row vectors forms a sentence embedding basis and all of the row vectors span the semantic key's entire commentary space. A semantic key is extracted from the input sequence during generation, and used to constructa comment sentence embedded from the external memory. Then the final comment is generated using the external memory embedding, as well as the post sequence embedding with a sequence-to - sequence model. The topics or the meanings of the statement can be interpretably controlled by manipulating the semantic keys.

Figure 5 describes the general context for the generation of checkable short-text conversations External Memory, information storage and retrieval. Initially, the system reads the user input andencodes it into a fixed-length vector with the use of semantic key mapper. First it stores semantickey in short-term memory and the remaining information is stored in external memory and producesuser response. Using addressing, the semantic key is applied to output. Finally, it provides anresponse to the question of the user.

3.7 General Framework

External Semantic Memory Driven Sequence-to-Sequence Learning system is composed of three components: encoder E, decoder D, and external Memory. The standard encoder-decoder system incorporates an external memory, which is supposed to be constructed using massive volumes of data. This research typically focuses on the basic semantic keys (e.g. keyword or topic) of a sentenceand then, given our target or context knowledge (memory), thinks about the relevant semantics and finally composes a answer sentence based on the newly written semantic output. Using the external semantic memory module we simulate this process. First get the output semantic key when generating a response, and address the memory blocks to get the output embedding. Finally, the decoder model produces the statement based on both the embedding of the input post and the embedding of the output phrase extracted from the external memory.

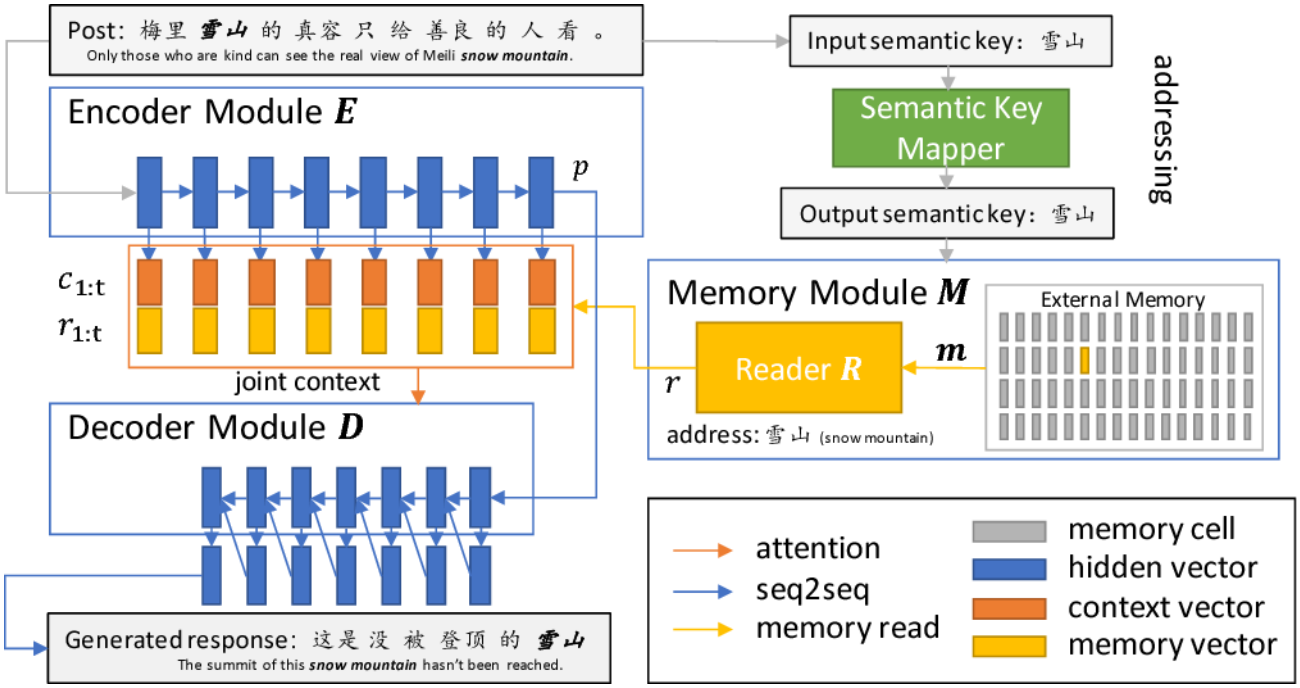


Figure 5.General Framework for the Generation of Controllable Short-Text Communicationwith External Memory

3.8 External Semantic Memory Construction

The content of each memory block is a matrix that includes embedded K representative statement sentence corresponding to a particular semantic output key. The "semantic key" may be represented by either a keyword (one-hot vector) or an embedding vector subject. The address of each block of memory is the index of the semantic key associated to the output. In the case of keyword-based representation, this is the single-hot vector keyword index. In the case of an embedding vector topic, the vector quantization is first applied to describe the embedding space topic. The semantic key index is the resulting code book index, then. Semantic output keys can be obtained by mapping the semantic input keys using the semantic key mapper S.

There are two different approaches used for constructing external memory data.

3.8.1 Encoder-decoder

The standard sequence-to - sequence learning (with or without attention) can be pre-trained given parallel STC data where both posts and comments are available. Here, data that vary from data for directed external semantic memory training in order to integrate external information, but data from similar corpus is preferred. Once the training has been completed, the decoder is discarded and only the encoder is used to convert all comments in the training data into embedding sentences. Ideally, want to encode the post-comment pair information and the link between posts and comments. We believe that using the encoder trained in encoding comments from post-comment pairs can capture the knowledge and relationship in some way. In addition, it is also consistent with the generation process because the encoder side uses the external semantic memory.

3.8.2 Auto-encoder

When the post-comment pairs are not available, large quantities of non-parallel sentences such as news, novel, or any other text materials can also be used to incorporate richer external knowledge. Here, the auto encoder is trained on non-parallel data set and the data is translated into embedded sentences.

3.9 Encoder-decoder External Semantic Memory Training:

External memory construction, the training procedure can be divided into two parts: pre-processing data and end-to - end encoder-decoder training.

3.9.1 Data Preprocessing:

In the memory construction, one sentence can be put in multiple semantic key classes. Also, the training data are processed under the same rule. Since the training corpus is post-comment pairs (i.e. input and output sentences), the training pairs are sorted according to their semantic keys. There are two advantages to this filtering: 1) more convenient for mini-batch training; 2) more likely to filter out some generic responses.

3.9.2 End-to-End Training and Generation:

The external semantic memory is fixed during training; hence, it is essentially an additional input to the decoder. The remaining model components are a standard carefully designed encoder-decoder structure. Choose one reply-side semantic key group from the training data at random, and then buildthe mini-batch. Note that its external memory address, i.e. relevant semantic comment-side key, was already known for each sample pair. The encoder-decoder target function is defined as

$$L = \sum_{t=1}^{T} \quad \log p(y_t | y_{1:t-1}, x, k(o))$$

One type of external memory device is the encoder-decoder model (ESED) that is controlled by external semantic memory. This model is used for constructing external memory: the simple sequence to sequence model (S2S) the sequence-to-sequence with attention (Atten) and an auto- encoder (AutoED) trained only on the comments (this can be seen as using non-parallel corpus). It is a 1-layer GRU with embedded word in 400 dimensions and concealed state vector in 800 dimensions. Nouns, adjectives and idioms are used as the keys to the semantic keys. With a uniform distribution between -0.05 and 0.05 all parameters are initialized. Adam optimizer[33] with an initiallearning rate of 0.0004 is used. The size of the Mini-Batch is set to 64. Replace the model HRED (Hierarchical Recurrent Encoder-Decoder) with LSTM, and use nouns and adjectives as keywords. All of these models are conditioned to achieve the best true perplexity in multiple epochs. During generation search for beams is used. It is worth noting that multiple semantic input keys are used in turn for the generation of multiple comment candidates for ESED.

INTERNATIONAL JOURNAL FOR INNOVATIVE RESEARCH IN MULTIDISCIPLINARY FIELD
ISSN(O): 2455-0620 [Impact Factor: 7.581]
Monthly, Peer-Reviewed, Refereed, Indexed Journal with IC Value : 86.87
Volume - 9, Issue - 10, October - 2023 Publication Date: 25/10/2023

4. SYSTEM ARCHITECTURE

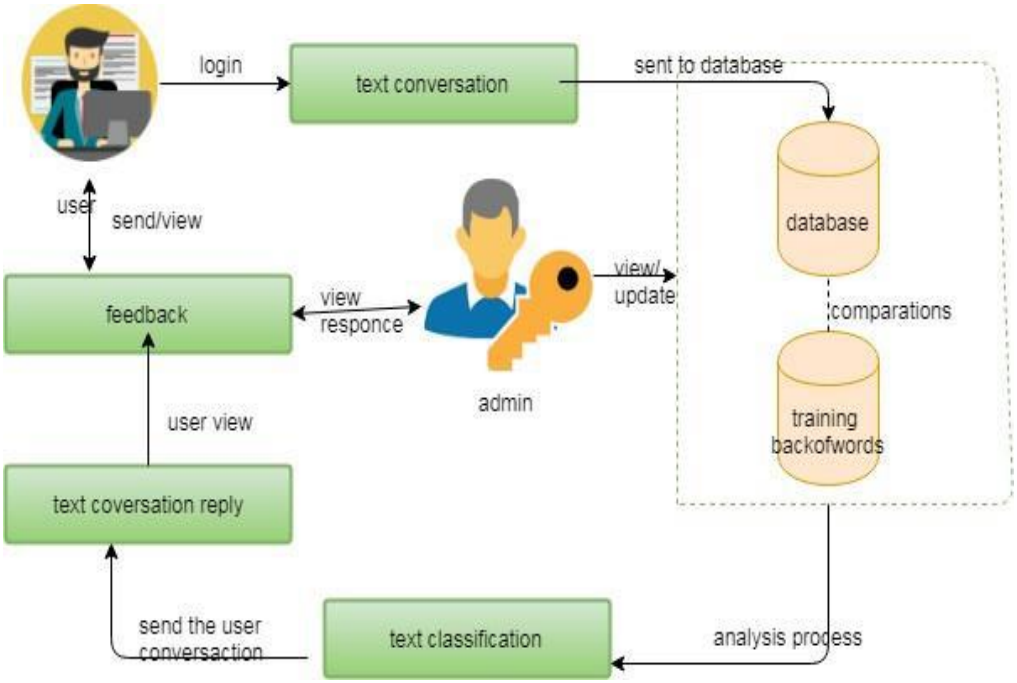


Figure 6. Architecture for Short Text Conversation

Figure 6 describes the present working machine architecture, in which it primarily includes twoessential phases of user operation and admin operation. User login to program initially by giving user name and password. The user may start conversation after logging in, which is stored in the database and handled by the admin. Also, the user sends reply to someone's post and see reply for his post and also sends feedback. Admin can do actions such as manually adding data to the database, viewing the responses and viewing the updates.

4.1 Training Dataset

User sends the text conversation which is stored in database and maintained by admin. The data in the training process aims to form the classification model. The testing process is the process to test the results of the classification based on the model that has been obtained. The first process is preprocessing data. The Preprocessing of the training data and testing is separately performed to simplify the further process. Preprocessing is carried out on the early stage before the process of training (using training data) and before the process of testing (using the testing data).

4.2 Text Classification

Data mining process or text classification is used for semantic reply**.** For short, text is sparse with a low number of features and doesn't provide enough word co-occurrence. That is why preprocessing the data is particularly important in short text classification; stop word removal (i.e. the removal of words that carry no useful information for the classification task) and stemming (i.e. the conversion of words into their *roots* or *stems*), are useful techniques for reducing the data deficiency and shrinking the feature space. Also, short text usually doesn't provide enough background information for a good similarity measure between different short texts. The main process in the training data set is comparing the user text conversation.

4.3 Process

The system generally consists of four phases: encoding, external memory, short term memory and decoding.

Encoder:

The recurrent neural network encoder decoder has become an effective and standard approach to sequence-to-sequence (seq2seq) prediction in general.

Usually the encoder part is a standard GRU, to map a sequence into a context vector of a fixed-size sentence level. The RNN encoder reads the sequence of inputs and generates a fixed-size context vector that represents a semantic summary of the input sequence. Initially, the user receives the inputand each time step accepts a single element of the input sequence, processes it, collects information for that element, and propagates it.

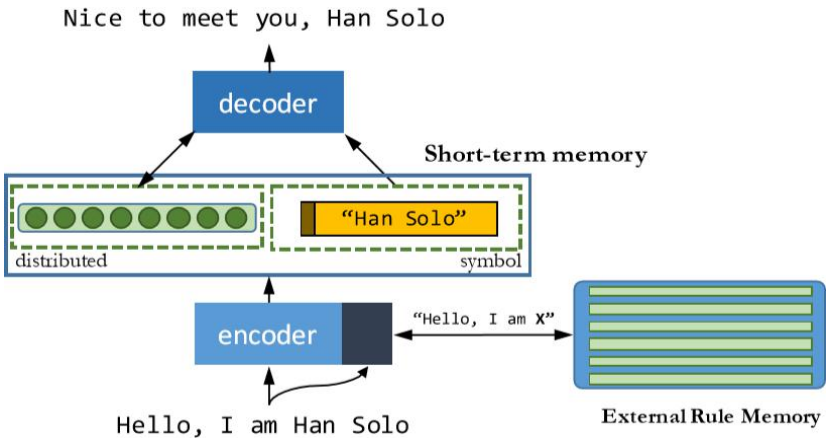


Figure 7. Example of Proposed System

Figure 7 explains the example of a proposed system in which, initially it receives input from user and encodes the input. It is divided into two forms: one is keyword and the other is general description. The description is to be stored in external memory. The key words are stored in short term memory. The short term memory consists of two parts: symbol and distributed. In symbol, it stores key words and in distributed, it stores both encoder and decoder data. Finally, it produces the reply.

External Rule memory:

External Rule memory is essential for the success of recurrent neural network based systems on many tasks such as question-answering, classification, and machine translation and reasoning. In all those models, the memory is used to store instance representations of multiple levels, analogous to "data" in computer, while the instructions are stored in the weights. The neural network and human experts can access this memory, and serve as an interface.

External memory is for the registration of rules on how to process input. The rule consists of pairing input and output series with choice of having common variable in them. The model first finds an application rule from its memory given an input sequence, and uses it to extract substring from the input using a pointer network. RNN then generates output sequence.

Short term memory:

Short term memory is a recurring artificial neural network used in deep learning. It is also referred toas active memory, it is the ability to hold data, but not to manipulate data, a small amount of information in an active mind, and it is readily available for a short time. The short-term memory is divided into two parts: distributed and symbol.

Distributed:

In distributed part, it can store both encoded and decoded information. It receives the input fromencoder and generates output in the form of decoder.

Symbol:

Short text conversation is used with semantic key to generate seq2seq generation. In short-term memory the semantic key can be stored. Short term memory is divided into two fields: symbol and description. The symbol will store semantic key in it adding to the final output this semantic keyword.

Decoder:

The context of the fixed- vector size can be provided as the Decoder RNN's initial state, or it can be connected to the hidden units at each time step. The number of Encoder and Decoder time steps involved need not be identical. A decoder network then uses this internal representation to output words until it has reached the end of the sequence token. Both the encoder and the decoder are used for LSTM networks.

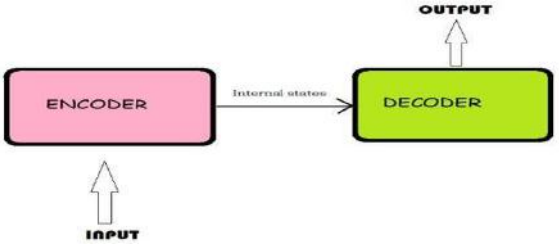


Figure 8. Encoding and Decoding process

Figure 8 explains the encoding decoding process, it receives input and encodes into fixed vector which is machine understandable language, decode the fixed length vector, generates reply and display the output.

5. METHODOLOGY :

5. 1 External Semantic Memory Guided Sequence-to-Sequence Learning algorithm:

The Directed Sequence-to - Sequence Learning System for External Semantic Memory consists of three components: an encoder E, a decoder D and an external Memory. The construction of external memory is independent of the encoder-decoder training. Generic gradient descend algorithm can be used to train the encoder-decoder model provided an external memory.

5.2 Encoder-decoder

Given parallel STC data, where both posts and comments are available, standard sequence-to - sequence learning can be pre-trained (with or without attention). Here, data can vary from data for directed external semantic memory training in order to integrate external information, but data from similar corpus is preferred. Once the training has been completed, the decoder is discarded and only the encoder is used to convert all comments in the training data into embedding sentences. Ideally, we want to encode the post-comment pair information and the link between posts and comments. The information and interaction can be captured in some way by using the encoder learned from post-comment pairs to encode comments. In addition, it is also consistent with the generation process because the encoder side uses the external semantic memory.

5.3 Algorithm:

- **Receive** an input post
- **Convert** post to a word embedding sequence $x_{1:T.}$
- **Extract** the input semantic key $k^{(i)}$
- **Map** it to output semantic key $k^{(0)} = s(k^{(i)})$
- **Find** associated memory block matrix m.
- **Encode** post using the encoder p, $c_{1:T}$= E($x_{1:T}$).
- **Read** memory context vector from the external semantic memory r = R(m,p).
- **Append** r to original encoder context vectors $\tilde{C}$= {[$c_{1;r}$],[$c_{2;r}$],...,[$c_{t;r}$]}.
- **Decode** to generate comment y = D(p,$\tilde{C}$).

5.4 Explanation

Let $x_{1:T}$ = {x_1,$x_{2,.....}$,x_T } represent the words embedded in the text, Where T represents the text length. The encoder module E, receives word embeddings, generates a dense representation p of the input post sentence and a set of context vectors,$c_{1:T}$= {c_1, $c_{2,....}$, c_T}. Every memory block's content is a matrix including K representative comment sentence embedding corresponding to a particular semantic output key. The "semantic key" may be represented by either a keyword (one-hot vector) or a topic embedding vector. The address of each block of memory is the index of the associated to the output semantic key. In case of keyword based representation, the one-hot keyword-based index is vector. In case of an embedding vector topic, the vector quantization is first applied to discrete the embedding space topic. The semantic key index is the resulting code book index, then semantic Output keys can be obtained by mapping the input semantic keys using the semantic key mapper S,

it is fed into a semantic key mapper S to convert to the semantic output key $k_{(0)}$

$$k^{(0)} = s(k^{(i)})$$

The semantic key mapper S can be trivial, e.g. using the semantic key directly on the input side asthe semantic key on the answer side,

$$\text{i.e.} k^{(0)} = k^{(i)}$$

After determining the semantic output key, must use it to index the external semantic memory M.M consists of memory blocks K, where K is the number of all possible semantic output keys. Each semantic output key corresponds to a block memory address, or index. D denotes the embedding dimension of a sentence. According to the process of memory construction (described below), D is equal to the dimensionality of the embedding of the input post. A memory reader R performs readingon the selected memory block m to construct an external memory background vector r

$$r = R(m,p)$$

Repeat it for each input word instance after obtaining the memory context vector r and add it to the original context c1: T to form a new collection of specific context vector C.

$$\tilde{C} = \{[c1;r],[c2;r],...,[ct;r]\}$$

During decoding the reply comment (referred to as y) is generated by the decoder D using the context vector from both the post sentence and the external semantic memory. The decoder is a uni- directional GRU with a specific background attention mechanism.

$$y = D(p,\tilde{C})$$

5.5 Recurrent Neural Network (RNN)

Recurrent Neural Network is commonly called Long-Term Memory (LSTM) network. Gated Recurrent Unit (GRU), map the input sequence to a fixed-dimensionality vector, and then another RNN to decode the vector target sequence.

$$r_t = \sigma(W_r x_t + U_r h_{t-1}) \quad (1)$$
$$z_t = \sigma(W_z x_t + U_z h_{t-1}) \quad (2)$$
$$\tilde{h}_t = \tanh(W x_t + U(r_t \odot h_{t-1})) \quad (3)$$
$$h_t = z_t \odot h_{t-1} + (1 - z_t) \odot \tilde{h}_t \quad (4)$$

where x_t is the input word embedding vector at time t, ht is the output hidden state vector, h_t is the output hidden state vector, rt is the output vector of the reset gate, z_t is the output vector of update

gate, W and U are weight matrices and $\odot$ denotes element-wise product. From the formula, GRU can take vector sequence of variable -length sequence $\{x1 \cdots xT\}$ from the formula and convert it to a fixed-length vector h_t. This recurring feature of the calculation makes it ideal for modeling sequences.

The encoder component is typically a regular GRU in sequence-to-sequence training to map sequence into a background vector at a fixed-size sentence level. Conversely, the decoder takes the context vector as an additional input and generates word-by-word output sequence.

Figure 9 explains the process of RNN in which it takes input and encode into fixed size vector by using Gated Recurrent Unit (GRU). GRU is also called as RNN and perform repeated action on input called as hidden layers after that it can store input in the form of fixed size context vector generator. After this, again perform RNN decoder and generate response to the input.

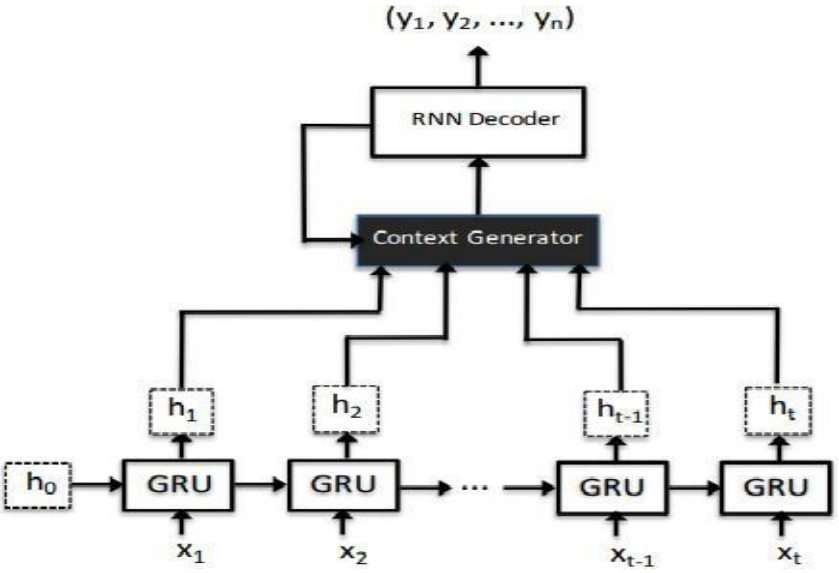


Figure 9. Converting Input into Encoder Format and Generating Output Using Decoder

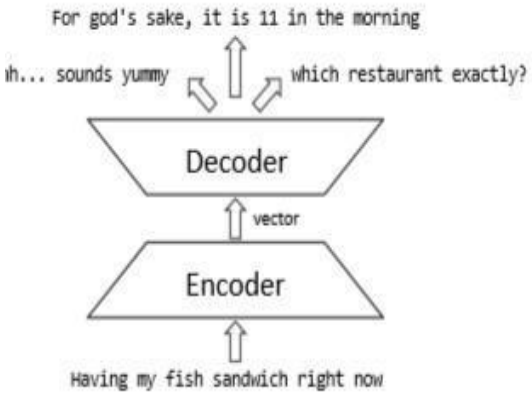


Figure 10. Example for Recurrent Neural Network Encoder and Decoder

The GRU decoder formula is similar to equations (1) to (3) except that the background variable is used as additional data. For Example, given $h_T^{(e)}$ as the output vector of the encoder, decoder equation (1) can be rewritten as:

$$r_t = \sigma(W_r x_t + U_r h_{t-1} + C_r c_t) \qquad (5)$$

Where C is the context vector's weight matrix and $c_t = h_T^{(e)}$ is the context vector at the sentence-level. Using attention mechanism, which replace the context vector $h_T^{(e)}$ with a time dependentcontext vector is commonly used extension of the encoder –decoder method. The time based context vector C_t is constructed using weighted sum of all encoder's hidden state vectors. The weights can be viewed as a distribution to describe the significance of all hidden state vectors in the encoder and decoders current hidden state vector. The context vector is calculated in many ways, and a popular implementation is used.

$$e_{ti} = h_t^{(d)\top} W_c h_i^{(e)} \quad \alpha_{ti} = \frac{\exp(e_{ti})}{\sum_{k=1}^{T} \exp(e_{tk})} \qquad (6)$$

$$c_t = \sum_{i=1}^{T} \alpha_{ti} h_i^{(e)} \qquad (7)$$

Where $h_t^{(d)}$ and $h_t^{(e)}$ the encoder and the decoder respectively hidden state vectors, T is thecumulative length of the encoder input sequence and is the final background vector at time t. Sequence-to - sequence learning with process of focus has been applied effectively to several tasks, such as questions and answers. This was extended to STC, short text communication. The framework for the encoder decoder tends to generate both short and dull responses. This phenomenon is due to a conversational nature. Unlike machine translation, where source sentence and target sentence convey the same meaning, where the summary text is simply a fragment of the source document, the answer to a post in STC will include additional details that does not appear in the post and involve an understanding of external knowledge. The additional details can be subjective, provided it is applicable to the article. Therefore, multiple suitable responses for one single post may exist. In comparison, target text for machine translation almost always differs on the surface level , i.e. lexicon or grammar.

The distribution of comments in the short text conversation is very complex and highly variable due to the semantic distinction. It is difficult for sequence to sequence model to capture the semantic mapping accurately from post to comment, as external information may be needed for mapping of this type. In addition, the cross-entropy training criterion, optimizing the KL divergence between true data distribution and model distribution, tends to fit all data distribution modes. This tendency makes the model distribution mode appear in a less likely region, when the model capacity is lower than the data distribution complexity. Meanwhile the perplexity reduction from a sequence-to - sequence model is smaller compared to a language model trained on statement. This phenomenon means the post might not provide sufficient information to generate the corresponding comment. Hence, general responses without substantiality can be highly likely and often generated.

To preserve coherence in longer responses, a self-attention mechanism is applied to the decoder, and an earlier stochastic beam-search algorithm with segment re ranking is implemented to inject diversity into the generation process. Although this kind of approach is successful in producing longer and informative answers, the question can not be eased. A general approach to solving dull response problem is to provide the decoder with additional information. Uses a topic model to extract specific post subject words. These extracted theme words are very important, and are very likely to appear in the answers. The decoder will focus on the additional subject words and is allowed to produce the additional subject words. In addition to additional information at word-level and exploit additional information at sentence-level, combining the generation model with a model based on retrieval. Retrieves the most similar post in the corpus, and takes the corresponding answer to the decoder as another input. Preserves important external

information about the person inside the memory network's communication past. The decoder then pays attention to the memory and selects the useful information to be produced automatically.

While due to the additional knowledge these methods improve the model and yield better results, these methods are not controllable. The variety of candidates within a beam is considered to be limited when beam search is used during decoding. Consequently, the candidates' generated responses of the above methods are generally similar, and differences are mostly at surface level. The underlying explanation for missing diversity is that, for the same article, such approaches neglect the fact that multiple suitable responses exist. When two people chat, the first person understands what the other person says first, then decides what to answer, and expresses it in natural language. The second stage includes external awareness such as common sense, awareness of the context and personal preferences. It is the most critical consideration for reaction diversity. An ideal communication structure would imitate the cycle of generation under control

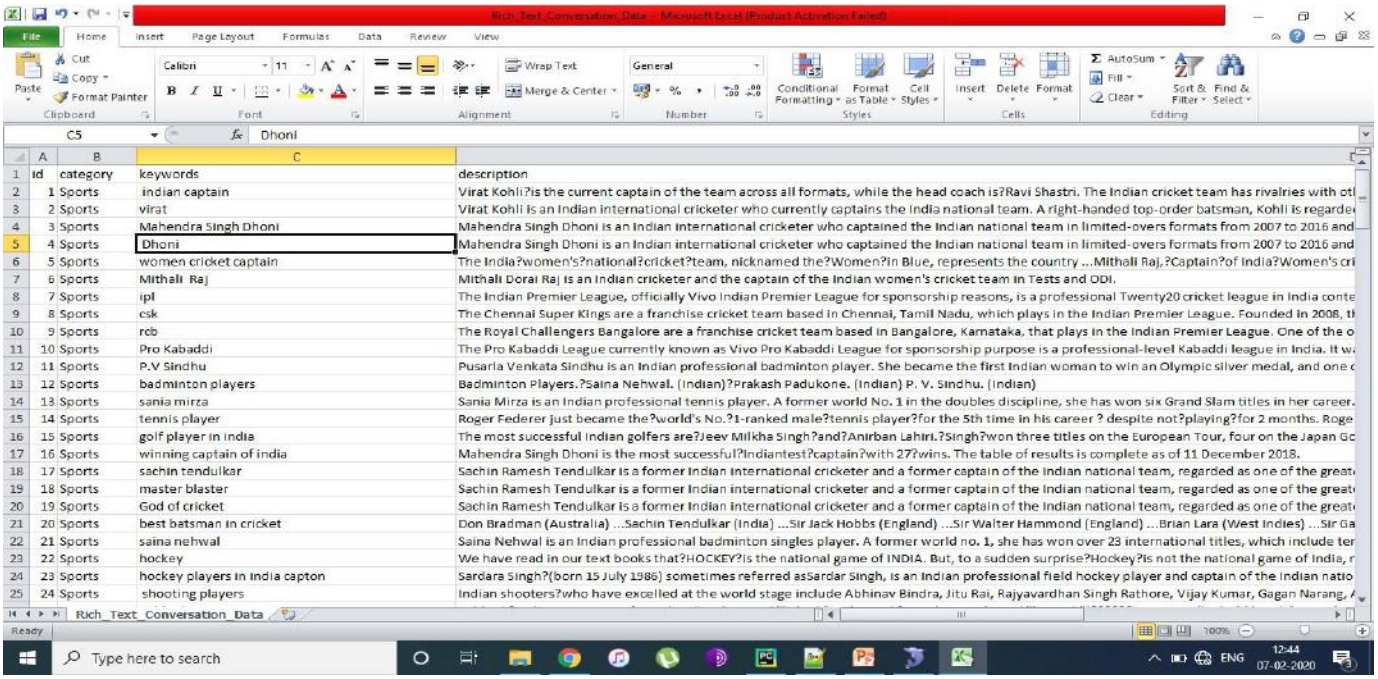
Some related works have been done addressing this issue. The method proposed, provided the article, retrieves few "facts" and then takes the supplementary materials for generation. The idea of introducing postage-related external data is very similar to the current work. However, this work focuses on the creation of external memory to allow efficient use of external data while focusing more on leveraging various variants of the multi-task method. However, they can regulate only the emotion or tense that is important in the STC mission, rather than the semantics.

Until producing the final responses, the semantics of the answer should be sought to mimic the phaseof conversation. For open domain conversation, however, there is no well-designed semantic representation. Adopt a two-step generation method, and consider keyword(s) as an alternate semantic response representation. First it generates a keyword according to point mutual knowledge, then it uses the method backward-forward to make the keyword appear in the answer. Proposes an implicit content introduction method which implicitly uses the auxiliary cue word information through a hierarchical gated fusion unit. The cue words are inserted into the cycle of generation but do not automatically appear in the reply. In comparison, it first generates a sequence of keywords, which is the noun sequence of the training response, then generates the final response according to the post and sequence of the nouns. The semantic representation in these models is the exact keyword(s). The depiction is very small. What we do care about is not the exact keywords, but the contents of the reply that are linked to those keywords. However, because multiple keyword sequences from beam search are still identical, these approaches can not address the diversity problem in full.

5.5 Data set

The Proposed system uses a short text conversation dataset in real time consisting of I d, category, keyword, and description functions.

Class includes information about the different categories: politics, films, education, sports and work related information, fashion, health care information. Keywords play an important role in this dataset, which contains essential knowledge. Based on this keyword, the user can extract informationbecause of this keyword, there is no need to check the entire dataset every time and generate correct response to the request of the user. The dataset contains the portion of the description in which the full information is available. Using semantic key, users can extract all the information they need.



id	category	keywords	description
1	Sports	indian captain	Virat Kohli?is the current captain of the team across all formats, while the head coach is?Ravi Shastri. The Indian cricket team has rivalries with ot
2	Sports	virat	Virat Kohli is an Indian international cricketer who currently captains the India national team. A right-handed top-order batsman, Kohli is regarde
3	Sports	Mahendra Singh Dhoni	Mahendra Singh Dhoni is an Indian international cricketer who captained the Indian national team in limited-overs formats from 2007 to 2016 and
4	Sports	Dhoni	Mahendra Singh Dhoni is an Indian international cricketer who captained the Indian national team in limited-overs formats from 2007 to 2016 and
5	Sports	women cricket captain	The India?women's?national?cricket?team, nicknamed the?Women?in Blue, represents the country ...Mithali Raj,?Captain?of India?Women's cri
6	Sports	Mithali Raj	Mithali Dorai Raj is an Indian cricketer and the captain of the Indian women's cricket team in Tests and ODI.
7	Sports	ipl	The Indian Premier League, officially Vivo Indian Premier League for sponsorship reasons, is a professional Twenty20 cricket league in India conte
8	Sports	csk	The Chennai Super Kings are a franchise cricket team based in Chennai, Tamil Nadu, which plays in the Indian Premier League. Founded in 2008, th
9	Sports	rcb	The Royal Challengers Bangalore are a franchise cricket team based in Bangalore, Karnataka, that plays in the Indian Premier League. One of the o
10	Sports	Pro Kabaddi	The Pro Kabaddi League currently known as Vivo Pro Kabaddi League for sponsorship purpose is a professional-level Kabaddi league in India. It w
11	Sports	P.V Sindhu	Pusarla Venkata Sindhu is an Indian professional badminton player. She became the first Indian woman to win an Olympic silver medal, and one
12	Sports	badminton players	Badminton Players.?Saina Nehwal. (Indian)?Prakash Padukone. (Indian) P. V. Sindhu. (Indian)
13	Sports	sania mirza	Sania Mirza is an Indian professional tennis player. A former world No. 1 in the doubles discipline, she has won six Grand Slam titles in her career.
14	Sports	tennis player	Roger Federer just became the?world's No.?1-ranked male?tennis player?for the 5th time in his career ? despite not?playing?for 2 months. Roge
15	Sports	golf player in india	The most successful Indian golfers are?Jeev Milkha Singh?and?Anirban Lahiri.?Singh?won three titles on the European Tour, four on the Japan Go
16	Sports	winning captain of india	Mahendra Singh Dhoni is the most successful?Indiantest?captain?with 27?wins. The table of results is complete as of 11 December 2018.
17	Sports	sachin tendulkar	Sachin Ramesh Tendulkar is a former Indian international cricketer and a former captain of the Indian national team, regarded as one of the great
18	Sports	master blaster	Sachin Ramesh Tendulkar is a former Indian international cricketer and a former captain of the Indian national team, regarded as one of the great
19	Sports	God of cricket	Sachin Ramesh Tendulkar is a former Indian international cricketer and a former captain of the Indian national team, regarded as one of the great
20	Sports	best batsman in cricket	Don Bradman (Australia) ...Sachin Tendulkar (India) ...Sir Jack Hobbs (England) ...Sir Walter Hammond (England) ...Brian Lara (West Indies) ...Sir Ga
21	Sports	saina nehwal	Saina Nehwal is an Indian professional badminton singles player. A former world no. 1, she has won over 23 international titles, which include ten
22	Sports	hockey	We have read in our text books that?HOCKEY?is the national game of INDIA. But, to a sudden surprise?Hockey?is not the national game of India, r
23	Sports	hockey players in india capton	Sardara Singh?(born 15 July 1986) sometimes referred asSardar Singh, is an Indian professional field hockey player and captain of the Indian natio
24	Sports	shooting players	Indian shooters?who have excelled at the world stage include Abhinav Bindra, Jitu Rai, Rajyavardhan Singh Rathore, Vijay Kumar, Gagan Narang, A

Figure 11. Short Text Conversation Dataset

5.6 Diversity and Substantiality Analysis

The system of work proposes two objective measures of analysis for STC: diversity and substantiality. The diversity and substantiality can be easily and directly calculated from the comments generated without knowing the commentary on the reference. This avoids the problemthat, in the comment text space, the coverage of references can be very small.

Diversity:

Reflects the richness of the words in the comments produced and defined as

$$\mathtt{Div} = \frac{\#(unique\ words)}{\#(all\ words)}$$

For all test sentences, the calculation is done by counting the total number of words and the number of uniquely appearing words. The underlying assumption here is that the richer the vocabulary generated, the more diverse the comments generated are. It can be easily seen from the metric; general answers without substantiality would reduce the metric for diversity.

Example for Diversity:

User enters the semantic key as virat, so virat sequence information is displayed as response.

Response:

Virat kohli is an Indian international cricketer who currently captains the Indian international team.Aright handed top order batsman, kohli is regarded as one of the best batsman in the world.

Diversity=7/19=0.36

Substantiality

Reflects the significant information contained in the replies provided, and is specified as the number of meaningful entities. The substantiality extracts entities by identifying, which is relatively limited, names of people, place names and names of organizations. Substantiality is the mean word number of the entity per sentence on the test set.

$$\mathtt{Sub} = \frac{\#(entity\ words)}{\#(sentences)}$$

Substantiality =3/4=0.75

The diversity and substantiality is achieved with external memory using RNN. Both the values ofdiversity and substantiality are below one, and also the values of diversity are less than substantial**.**

6. CONCLUSIONAND FUTURE ENHANCEMENTS :

This approach introduces a novel method for improving short text conversation tasks. By integrating external semantic memory into the encoder-decoder framework (RNN), we effectively address issues related to vague and uninformative responses, resulting in more varied and concrete answers. This separation of external memory construction and neural network training permits the use of non-parallel datasets, expanding its applicability. Furthermore, our approach allows for the fine-tuning of response semantics by manipulating the semantic key mapper, offering a fresh approach to generating richer responses. The combination of external memory with neural networks ensures precise semantic responses and reduces system processing time. Looking ahead, the challenges of short text communication can be further overcome by exploring other deep learning and data mining techniques. Methods employing RNNs, CNNs, and external memory guided sequence-to-sequence learning may provide more efficient processing, offering precise semantic replies in less time. Utilizing high-quality datasets will also contribute to achieving more accurate results, and the exploration of various deep learning algorithms can yield more substantial and diverse responses to input posts.

REFERENCES :

1. Makoto P Kato, Kazuaki Kishida, Noriko Kando, Tetsuya Saka, and Mark Sanderson, “Report on ntcir-12: The twelfth round of nii testbeds and community for information access research,” in ACM SIGIR Forum. ACM, 2017, vol. 50, pp. 18–27.
2. Zongcheng Ji, Zhengdong Lu, and Hang Li, “An information retrieval approach to short text conversation,” arXiv preprint arXiv:1408.6988, 2014.
3. Lifeng Shang, Zhengdong Lu, and Hang Li, “Neural responding machine for short-text conversation,” arXiv preprint arXiv:1503.02364, 2015.
4. Oriol Vinyals and Quoc Le, “A neural conversational model,” arXiv preprint arXiv:1506.05869, 2015.

5. Ilya Sutskever, Oriol Vinyals, and Quoc V Le, "Sequence to sequence learning with neural networks," in Advances in neural information processing systems, 2014, pp. 3104– 3112.
6. Kyunghyun Cho, Bart Van Merrienboer, CaglarGulcehre, Dzmitry Bah- ¨ danau, FethiBougares, Holger Schwenk, and YoshuaBengio, "Learning phrase representations usingrnn encoder-decoder for statistical machine translation," arXiv preprint arXiv:1406.1078, 2014.
7. DzmitryBahdanau, Kyunghyun Cho, and YoshuaBengio, "Neural machine translation by jointly learning to align and translate," arXiv preprint arXiv:1409.0473, 2014.
8. Alexander M Rush, Sumit Chopra, and Jason Weston, "A neural attention model for abstractive sentence summarization," arXiv preprint arXiv:1509.00685, 2015.SubhashiniVenugopalan, Marcus Rohrbach, Jeffrey Donahue, Raymond Mooney, Trevor Darrell, and Kate Saenko, "Sequence to sequencevideo to text," in Proceedings of the IEEE international conference on computer vision, 2015, pp. 4534–4542.
9. Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan, "A diversity- promoting objective function for neural conversation models," arXiv preprint arXiv:1510.03055, 2015.
10. Alessandro Sordoni, Michel Galley, Michael Auli, Chris Brockett, Yangfeng Ji, Margaret Mitchell, Jian-Yun Nie, Jianfeng Gao, and Bill Dolan, "A neural network approach to context-sensitive generation of conversational responses," arXiv preprint arXiv:1506.06714, 2015.
11. Sepp Hochreiter and J¨urgenSchmidhuber, "Long short-term memory," Neural computation, vol. 9, no. 8, pp. 1735–1780, 1997.
12. Junyoung Chung, CaglarGulcehre, KyungHyun Cho, and YoshuaBengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," arXiv preprint arXiv:1412.3555, 2014.
13. Minh-Thang Luong, Hieu Pham, and Christopher D Manning, "Effective approaches to attention-based neural machine translation," arXiv preprint arXiv:1508.04025, 2015.
14. Jun Yin, Xin Jiang, Zhengdong Lu, Lifeng Shang, Hang Li, and Xiaoming Li, "Neural generative question answering," arXiv preprint arXiv:1512.01337, 2015.
15. Yuanlong Shao, Stephan Gouws, Denny Britz, Anna Goldie, Brian Strope, and Ray Kurzweil, "Generating high-quality and informative conversation responses with sequence-to-sequence models," in Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, 2017, pp. 2210–2219.
16. Chen Xing, Wei Wu, Yu Wu, Jie Liu, Yalou Huang, Ming Zhou, and Wei-Ying Ma, "Topic aware neural response generation.," in AAAI, 2017, pp. 3351–3357.
17. Yiping Song, Rui Yan, Xiang Li, Dongyan Zhao, and Ming Zhang, "Two are better than one: An ensemble of retrieval- and generationbased dialog systems," arXiv preprint arXiv:1610.07149, 2016.
18. [conversation model," arXiv preprint arXiv:1702.01932, 2017.
19. Hao19] MarjanGhazvininejad, Chris Brockett, Ming-Wei Chang, Bill Dolan, JianfengGao, Wen-tau Yih, and Michel Galley, "A knowledge-grounded neural
20. Zhou, Minlie Huang, Tianyang Zhang, Xiaoyan Zhu, and Bing Liu, "Emotional chatting machine: Emotional conversation generation with internal and external memory," arXiv preprint arXiv:1704.01074, 2017.
21. Zhiting Hu, Zichao Yang, Xiaodan Liang, RuslanSalakhutdinov, and Eric P Xing, "Controllable text generation," arXiv preprint arXiv:1703.00955, 2017.