Computers, Materials & Continua
DOI:10.32604/cmc.2022.021033
Article

Tech Science Press

BERT for Conversational Question Answering Systems Using Semantic Similarity Estimation

Abdulaziz Al-Besher[1], Kailash Kumar[1], M. Sangeetha[2,*] and Tinashe Butsa[3]

[1]College of Computing and Informatics, Saudi Electronic University, Riyadh, 11673, Kingdom of Saudi Arabia
[2]Department of Information Technology, SRM Institute of Science and Technology, Kattankulathur, India
[3]Department of Information Technology, Harare Institute of Technology, Belvedere, Harare
[*]Corresponding Author: M. Sangeetha. Email: sangeetk@srmist.edu.in
Received: 20 June 2021; Accepted: 27 July 2021

Abstract: Most of the questions from users lack the context needed to thoroughly understand the problem at hand, thus making the questions impossible to answer. Semantic Similarity Estimation is based on relating user's questions to the context from previous Conversational Search Systems (CSS) to provide answers without requesting the user's context. It imposes constraints on the time needed to produce an answer for the user. The proposed model enables the use of contextual data associated with previous Conversational Searches (CS). While receiving a question in a new conversational search, the model determines the question that refers to more past CS. The model then infers past contextual data related to the given question and predicts an answer based on the context inferred without engaging in multi-turn interactions or requesting additional data from the user for context. This model shows the ability to use the limited information in user queries for best context inferences based on Closed-Domain-based CS and Bidirectional Encoder Representations from Transformers for textual representations.

Keywords: Semantic similarity estimation; conversational search; multi-turn interactions; context inference; BERT; user intent

1 Introduction

Conversational search is one of the most critical areas in Natural Language Processing (NLP); hence, researchers' ambition is to understand user intent in multi-turn conversations to simulate human-to-human interaction in Conversational Assistants (CA). CSS can be defined as an approach to find information in a multi-turn conversation, and it has long been associated with Information retrieval systems. The adoption of CA in Conversational Search Systems (CSS) is currently rising, which has attracted much attention from researchers. The most common framework for CA mainly focuses on Natural Language Understanding (NLU) [1] to design and develop systems that can better understand human language. The objective is to understand NLP and identify the informational users' needs (user intent) from natural language by analysing textual information.

In any CSS, the critical task is understanding user intent from utterances of that conversation [2]. Recent NLP models such as Bidirectional Encoder Representations from Transformers (BERT) [3], Robustly Optimized BERT-Pretraining Approach (RoBERTa), Generalized Autoregressive Pretraining for Language Understanding (XLNet), and Generative Pretrained Transformer 2 (GPT-2), have been outperforming humans on the competition datasets, Stanford Question Answering Dataset (SQUAD), and General Language Understanding Evaluation benchmark (GLUE). These advancements created much interest in conversational search regarding concepts such as the ability to identify user intent from utterances and also the ability to provide answers or solutions based on user's questions. Tasks involving questioning and answering in multi-turn environments using datasets like SQUAD would be best solved using models like BERT. SSE and context inference allow the model to deal with partial, limited, or incomplete questions from users who do not know how to express their informational needs and also with questions that Chabot designers did not expect though searching the whole knowledge base for an answer based on the question similarity [4]. There is a strong belief that CSS and ConvQA should provide helpful information by getting limited information from the user. Most CSS are not capable of understanding input with partial information as well as input with multiple turns. The ambiguous nature of questions from users will often require additional information for clarification, which often creates a challenge in ConvQA. The essential aspects of conversational search in question answering determine User Intent, NLU in multi-turn interactions, and CA.

Determining user intent is one of the key features for question answering in multi-turn conversations. In multi-turn interactions, the intent represented by the initial question of the user determines the flow of the conversation and how CA will process later utterances. Modelling multi-turn interactions between CA and users requires accurately identifying user intent [5]. A single conversational search session can consist of several utterances, with each utterance representing different user intent. Determining user intent in such scenarios becomes a challenge to provide the most suitable answer to the user. Depending on the NLP task, there are many classifications for user intent. Some latest research identifies user intent as the classification of statements in multi-turn contexts [6], for example, the user's initial quotation (question) is classified as the Origin Question (OQ), and utterances that represent additional data from the user are classified as Further Details. Other functions define user intent as the user intention is referenced in an utterance; for example, the utterance "I would like to buy a laptop" denotes the means to purchase [7]. For this research, the representation of user intent is the information the user intends to get as a response to a given question.

NLU refers to extracting meaning from natural language [8]. Input from users is not always straightforward. Most of the input from users is non-factoid and will often trigger multi-turn interactions between the agent and the user. For example, a query "How do we upgrade to Windows 10?" does not contain all the information needed to provide the most appropriate answer and requires that the user provide additional information to get the most suitable response or answer. Most often, CAs have to keep track of the change of user intent throughout the conversation, which is a challenging task, especially for conversations with several turns. Previous research helps resolve NLU challenges for single utterances to extract user intent and classify core features called slots from that single utterance using slot filling [9]. To understand the informational needs of the user from such conversations, more complex NLU techniques are required. CA should understand the context of each utterance in the conversation, elicit relevant knowledge in case of continuous evaluation of the user's informational demand, and enhance previous answers to improve the present answers. This modeling contextual representations from past conversations and inferring

them from users based on some similarity algorithm will help to determine user intent more accurately and quickly. It will, in turn, eliminate the number of turns needed to understand the users' informational needs.

The potential of CA to simulate human conversations in their natural forms, such as text or speech, enhances Semantic Similarity Estimation (SSE). Simulating human conversations should allow Question Answering Chatbots to provide the most accurate user questions [10]. It can be achieved by analyzing and identifying specific keywords and phrases from both text and speech. By focusing on conversation flow, CA should analyze the contextual data of the conversation to learn the relationships between words and utterances for processing answers. Utterances within a conversation usually represent different intent types, and by analyzing these utterances, CA will understand the user's intent. CA must be trained using a large domain knowledge base for higher quality language understanding in multi-turn conversations. Training CA using a variation of conversations with different informational needs should improve the performance of CA on question answering. CA is based on several key aspects: mode of interaction, CA usage, modeling techniques used, and the knowledge base or domain. By considering the mentioned aspects, CA will determine the contextual conversation data used for identifying the user's informational needs through NLU [11]. Emulating how people look for information regarding asked questions requires understanding the two types of domains related to CSS and ConvQA systems, namely Closed-Domain System (CDS) and Open-Domain System (ODS).

In CDS conversational search, the questions are limited to predefined domains and domain knowledge (e.g., tech support questions based only on Microsoft products); generally, CSS should answer a wide variety of user questions using contextual data from different domains to find answers. CDS Conversational Search Systems find information based on context from a predefined domain since they are trained to answer domain-specific questions. Since ODS Conversational Search Systems are limited to answering specific domains, Researchers focus on search systems that can answer different user questions. ODS Conversational Search Systems can generalize questions from users to answering questions from different domains. ODS can use different domain-based contexts from different knowledge bases to meet the user's informational needs and provide the most accurate answers. ODS can be helpful, especially when users don't know the particular domain to which their question is related. The main challenge of such systems narrows down candidate context for question answering; arriving at the answer may be a constraint on the time efficiency of the model in providing the answer to a user.

CSS features natural conversations with users. The generation of responses is mainly based on the level of confidence obtained from the context provided by users, and the sequence of dialogue contexts is considered for information finding. Interactions between users and CA can be divided into two classes: single and multi-turn interactions, as illustrated in Fig. 1. Single-turn interactions provide answers based on the immediate user question (utterance) and do not require additional information to answer the question (i.e., single utterance just before the answer). On the other hand, multi-turn interactions generate a response based on multiple interactions between the user and the system. Utilizing SSE for question-context mapping in CSS and ConvQA systems allow CAs to figure out the user's informational needs before recommending an answer. A typical CSS is one in which the user initiates the conversation with an intent-based question. The system will ask for additional information through follow-up questions to understand the user's informational needs. When the system is confident enough, it will then suggest or retrieve the appropriate information to the user. Furthermore, the system will retrieve the answer iteratively throughout the interaction process, where it takes more than 2 turns for the agent to understand the user's

informational needs and generate the appropriate answer. This form of multi-turn interaction opens up new possibilities for CSS.

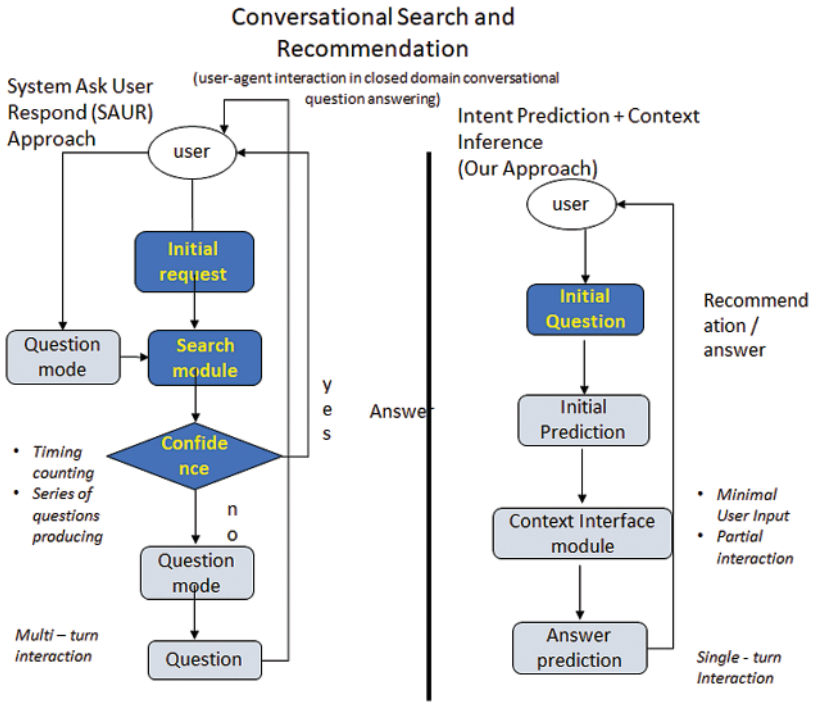

Figure 1: User-agent interactions in CSS

SSE in CSS allows a system to understand user intent without engaging in multiple rounds of message exchanges. Instead of asking for additional clarification, the system will infer conversational context to the user's question based on similarity computations. The model will leverage the recent question from the user and intent to look into conversational search to provide an answer as a recommendation with minimal user input and incomplete details. Each question from the user relies on inferring context from a single session to connect intent with the question. This work aims to understand user intent by utilizing BERT contextual representations for SSE to infer past conversation context on the current question using limited information from the user. Given a recent question from the user, this system wants to understand the user intent by computing the semantic similarity between the representations of the current question with the contextual representations of the previous conversations and then inferring detailed contextual information to the question at hand. User intent can be defined in many ways in the field of CS. This framework describes the user's intention to obtain information for a particular question. Predicting user intent comes from the need to understand the user's informational needs to provide the most accurate answer that meets the user's needs without additional user input. Like humans, CSS must learn to identify closely related or highly similar questions to refer historical context based on the question similarity. By referring to the historical conversational context, the system will understand the user's informational needs without carrying out the same process of requesting additional information from the user for clarifications, especially when it comes to similar questions. This approach helps to provide fast solutions with minimal user input. We utilize BERT for language

representations and understanding because of its ability to understand long-term dependencies in large text. Bert is a *state-of-the-art* NLP model for language representation from Google Artificial Language [12].

BERT uses bidirectionality by pre-training on mask language modeling and next sentence prediction, making it suitable for achieving the best contextual representations of each conversation for language understanding. BERT for ConvQA works and performs excellent on a relatively large number of words, making it suitable for understanding multi-turn interactions in CSS. Since BERT for ConvQA is trained on SQUAD data, summary paragraphs and related questions and not multi-turn interactions for dialogues. This model aims to construct BERT for intent prediction and question answering to understand the language in a multi-turn environment, which typically involves several turns. We conduct our experiments on predicting user intent CSS using the MSDialog data [13]. The data contain interactive dialogues between users seeking technical support and answer providers. Most user questions are often non-factoid and require further conversational interactions to build a solid understanding of the user's needs. The answer providers are, in this case, the Microsoft stuff and some experienced Microsoft product users (human agents). The answer (user intent) is the user's intent to get a question related to Microsoft products.

2 Related Works

Several CSS and NLU advancements have created new CSS and ConvQA systems' research interests over the years. Despite these advancements, understanding the nature of conversational search is still a difficult task. There remains a challenge of understanding the user's informational needs (user intent) in an interactive environment. The focus of ConvQA is to model change in user intent in multi-turn interactions. The intuition is based on handling conversation history between cycles in a multi-turn environment. This is achieved by selecting a subset of past turns (previous answers) based on the level of importance using a rule-based method. The model then embeds past answers for ConvQA. Given a question q_t from the user, history modeling expects the agent to refer to the previous answer a_{t-1} for q_{t-1} to understand the informational needs of the recent question from the user. The critical aspect of ConvQA is on using history turns for understanding the informational needs of the user. ConvQA performs history answer embedding to the recent question for a given informal session to understand the user's informational needs. History answer embedding allows the model to understand the user's intent through conversation history modeling for a particular conversational session. Combining earlier answers with the recent question from the user enables the agent to determine user intent.

The ConvQA method is suitable for understanding intent based on previous utterances within a particular conversation session. However, this approach is associated with multi-turn interaction, which constraints the time complexity of the model for generating an answer. Furthermore, in a multi-turn setting, using the sequential order of question-answer sets to understand user intent may have a detrimental impact on the CSS and ConvQA systems because user intent tends to change from one turn to the next. In such scenarios, understanding user intent for answer generation becomes difficult. For the same question from a different user, the ConvQA system may again go through several multi-turn interactions to understand user intent, making the process redundant, seeing that an answer for that same question was already generated in the previous session. The BERT system approach focuses on inferring past conversational context to the current user question based on some degree of similarity between the current question and the context of past CS. The contextual conversation data is modeled using BERT's next sentence prediction task. By inferring context from previous similar conversations to the current question,

this model understands the user's informational needs and provides answers without requiring additional information. The approach performs conversation contextual data modeling, which indirectly deals with the unexpected changes in user intent. Context modeling is performed based on the intent represented by the original question of conversation c_i. By focusing on utterances representing the same intent as that of the original question, this model infers the most accurate past conversational context to the question.

Existing approaches use a System Ask–User Respond (SAUR) approach for CS [14]. Naturally, people engage in multi-turn interactions when seeking information. SAUR aims to comprehend user's requirements by fetching answers based on user feedback. According to SAUR, processes that can start answering appropriate questions dynamically can better understand user needs, which is one of the essential aims of CSS and ConvQA. SAUR integrates sequential modeling and concern via multi-memory network architecture and an individualized version for CS and recommendation. This approach to CS and recommendation focuses on feature sets for the CS to manage and control user acceptance to comprehend user needs. However, this presents a scenario in which a user is given a practically identical question to ask questions historically; the system rehashes the same process of asking the learners to identify the user's informational needs rather than relating the user to related research conversations. Also, the user may ask follow-up questions that do not represent the same intent as the previous utterances, and this will start a new search altogether. ConvQA suggests that to understand the current information needs of the user, the model should be able to handle the conversation history of the current conversational search session. The approach used in this system will be capable of understanding user intent through context inference based on question similarity, and from the inferred context, we can determine or predict the user's informational needs.

Some methodologies to SSE used RL in User Chat Bots; the task of SSE is addressed as a task to assume relevant questions that users might be interested in [15,16]. The approach models SSE as a Markov decision process and implements Reinforcement Learning (RL) to find the best recommendation methods. Unlike other existing techniques, which predict the list of items likely to be of interest to users by depending on the immediate benefit rather than the long-term benefits of each recommendation, the analysis proved to review the inter-relationships between the user dynamics and recommends questions using a *N-Step* sequential decision process. The model will suggest and add a sensible question to the recommendation list at each turn. The model helps to understand clicks and user satisfaction by resetting its ranking results of the top *'N-Step'* recommendations based on user behaviour patterns and question popularity. The approach demonstrates the SSE task by generating better guidelines.

The approach using attentive history selection for question answering in CSS introduces a history attention mechanism to select conversation histories based on attention weights to understand and answer the question called *"Soft Selection."* For each turn in a conversation, different weights are allocated based on their usefulness in answering the current question of the user. Applying attention weights to utterances within a conversion allows the model to capture the importance of history turns. Furthermore, the method incorporates the position information for history to examine the importance of turn position in conversation history modeling. This work realizes the need to learn to answer current questions based on past conversations to limit the interactive process to a single turn. Another related yet different approach is Neural Re-entry Prediction Combining Context and User History. It uses a neural network framework that focuses on re-entry prediction given in a conversation by combining context and user history. The model learns meaningful representations from both the conversation context and user history to predict

whether they will return to a conversation they once participated in. The paper illustrates the importance of historical conversational context in understanding user utterances. This approach focuses on utilizing BERT context, representations for conversation context modeling and SSE. This model focuses on the conversational context of past conversations, and the similarity of user questions is an essential aspect for CSS and ConvQA.

3 Proposed Methodology

3.1 Problem Statement

Given a question q_i from the user, the task is to relate q_i with CS from past sessions to find and infer past conversational context c_i^k to q_i based on the highest semantic similarity score for question understanding and then generate the answer a_i to q_i, where c^k is the i^{th} conversation consisting of 'k' utterances after data modeling. Fig. 2 shows the system flow of this model approach.

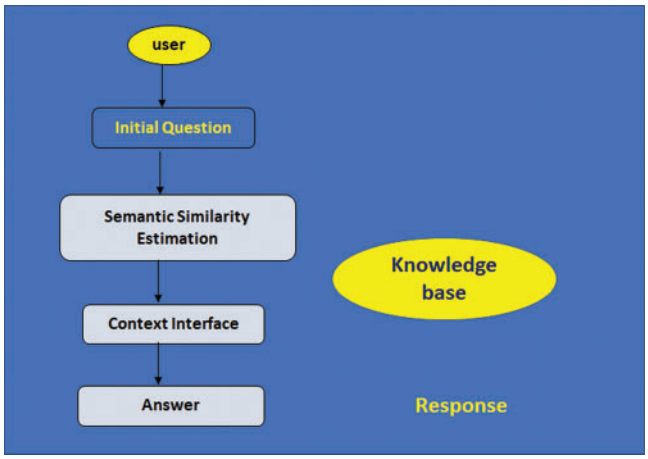

Figure 2: The flow of SSE, CSS, and Q&A

3.2 Overview

3.2.1 BERT Encoding for Intent Semantic Similarity Estimation and Question Answering

This approach utilizes the BERT model to encode the question q_i and the inferred conversation context 'c', into contextualized text representations for SSE. BERT is a cutting-edge, pre-trained language model for NLU that employs transformers to learn deep bidirectional representations. Given a training instance (q_i, c), pair the question and the conversation context into a single sequence. The input sequences are fed into the BERT encoder, and BERT generates contextualized representations for each sequence based on the token, segment, and position embedding. BERT is well suited for understanding the given textual information and deriving answers from the text. To understand the textual information for question answering, BERT was trained on the Stanford Question Answering Dataset (SQUAD), consisting of questions with a span of text from that particular textual data. The BERT model for SSE and ConvQA converts the MSDialog data structure to that of SQUAD data. Utterances from each conversation will be treated as contextual information for that particular conversation. The contextual information will provide the BERT Model with the features needed to understand the context-related question.

BERT model for ConvQA is limited to text not longer than 512 *tokens per sequence*. It makes it suitable for dealing with long sequential data from multi-turn interactions associated with each conversation. When the sequential data exceeds 512 *tokens per sequence*, understanding the data becomes a challenge.

3.2.2 Semantic Similarity Estimation in the Conversational Question Answering Framework

The system presents a modularized design framework for SSE in ConvQA as an abstract in Fig. 3. The framework mainly focuses on three key components: SSE (for determining user intent), context inference, and answer prediction or generation. Given a training instance (c_i^k, q_i, a_i), the SSE module chooses the conversation context c_i^k that is semantically similar to the given question q_i. The selected context is related to the model, which then learns the start and end vectors of the answer span from the inferred conversational context. It is based on the intuition that highly similar questions often go through the same context to understand the user's informational needs. Here, the conversational contextual data modeling and SSE model implementations are introduced in the following sections. In this research, the model employs a primary method as the conversational context inference in which the most relevant conversational context based on semantic similarity is inferred to the current question of the user for intent prediction. It is based on the intuition that similar questions often result in the same answer or solution, so instead of asking the user the same questions for clarification, this process minimizes user input and infer past related conversational context to the current question.

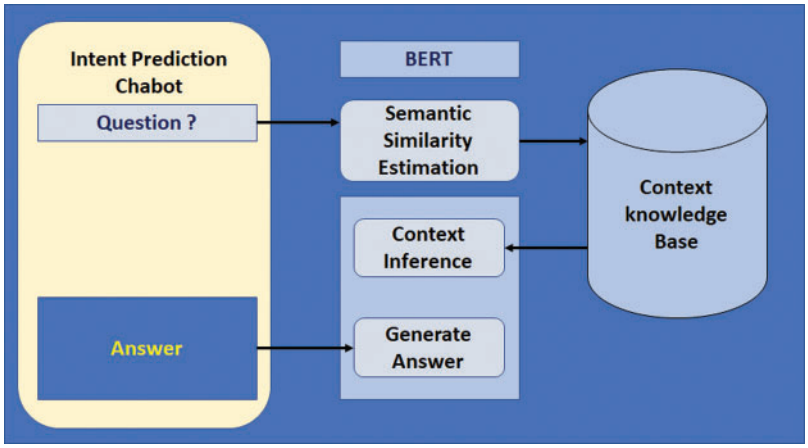

Figure 3: Framework for BERT Model

3.2.3 Semantic Similarity Estimation and Context Inference

Given a question from the user, the model performs SSE by comparing the similarity of the question with the contextual representations of each past CS in the knowledge base. By converting text (question and the contextual data) into Term Frequency-Inverse Document Frequency (TF-IDF) vectors, we compute the cosine similarity between the question utterance and each contextual conversation data in the knowledge base, as shown in Fig. 4. Similarity determines how close or related the given question q_i is to each conversation context c_i in the knowledge base in terms of meaning or context. The question is represented into a vector form, whereas the contextual data are represented in matrix form (e.g., TF-IDF): $tf - idf(t,c)tf(t,c) \times idf(t)$. The

Cosine similarity of the question and the conversation contextual data ranges from 0 to 1, where the score of 1 means that two vectors are highly similar. Eq. (1) is the Cosine similarity for this context inference module between two non-zero vectors.

$$Similarity = \cos^{-1}\theta = \frac{q.c}{\|q.c\|} = \frac{\sum_{i=1}^{n} q_i c_i}{\sqrt{\sum_{i=1}^{n} q_i^2}\sqrt{\sum_{i=1}^{n} c_i^2}} \tag{1}$$

SSE and context inference are based on the cosine similarity between the question TF-IDF vector and the conversational context TF-IDF features. This similarity function allows the model to rank the conversation contexts in the knowledge base and infer the context with the highest score to the question posed by the user. After SSE (selecting the most relevant conversation contextual data), the model infers that particular data to the question utterance and sends them to the question-answering module.

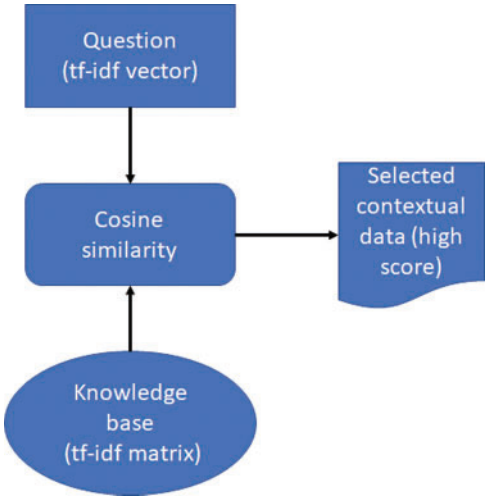

Figure 4: SSE and conversation context inference module

3.2.4 Question Answering

An essential aspect of finding the answer to the given question lies in inferring the most relevant conversational context to the given question and using that context to determine user intent. From the inferred contextual data, this model can then predict the answer text span. For example, given a user question q_i and the inferred conversational context c_i^k, here used a different approach to find the answer. For the training model, the input is the original question of the conversation c_i and the modeled utterances of the same conversation as context, the output is the probability of context tokens being the start or end tokens of the answer span. The model finds the most probable answer from the inferred contextual data by computing the START/END probabilities of the answers. For each contextual data in the knowledge base, the likelihood is computed for START/END tokens of the answer span based on the Softmax function regarding the given question. Given the word i, and its hidden vector T_i, the likelihood of the word being

the START/END of the answer span (P_i^S, P_i^E) is computed as, Eq. (2)

$$P_i^S = \frac{Exp^{S.T_i}}{\sum_j^i Exp^{S.T_j}}, \quad P_i^E = \frac{Exp^{E.T_i}}{\sum_j^i Exp^{T_j}} \tag{2}$$

The task is to predict the answer using the inferred context. As shown in Fig. 5, in the answer prediction task, the model represents the input question and inferred context as a single paired sequence, with the current question of the user using the **Q** embedding and the inferred context using the **C** embedding. The models represent the final hidden vector for the input token i as $T_i \in H$ and introduce a start vector $S \in H$ and end vector$\in H$. The dot product between $\boldsymbol{T_i}$ and $\boldsymbol{S}$ is used to calculate the probability of word 'i' being the start of the answer text span followed by a Softmax over all of the words in the inferred context of Eq. (3)

$$P_i^S = \frac{Exp^{S.T_i}}{\sum_j Exp^{S.T_j}} \tag{3}$$

The same formula is used for computing the end of the answer text span, Eq. (4)

$$P_i^E = \frac{Exp^{E.T_i}}{\sum_j Exp^{E.T_j}} \tag{4}$$

The score of a candidate span from position 'i' to position 'j' is defined as, Eq. (5)

$$S.T_i + E.T_j \tag{5}$$

and for prediction, this model uses the maximum scoring text span where $j \geq i$. The result is the sum of the likelihood of the correct START/END vectors. In the model architecture illustrated in Fig. 5, the question is mapped, the conversational context is packed, and the resulting sequence is fed to this model, and then a representation is also generated for each token on the token, segment, and position embedding. Next, a vector representation for the START/END position is learned. It will be used to compute the answer span based on the given question. The loss will be computed as the average of the cross-entropy loss for the START/END positions. The model should then produce the following interactive interface showing output for the given question based on past conversations.

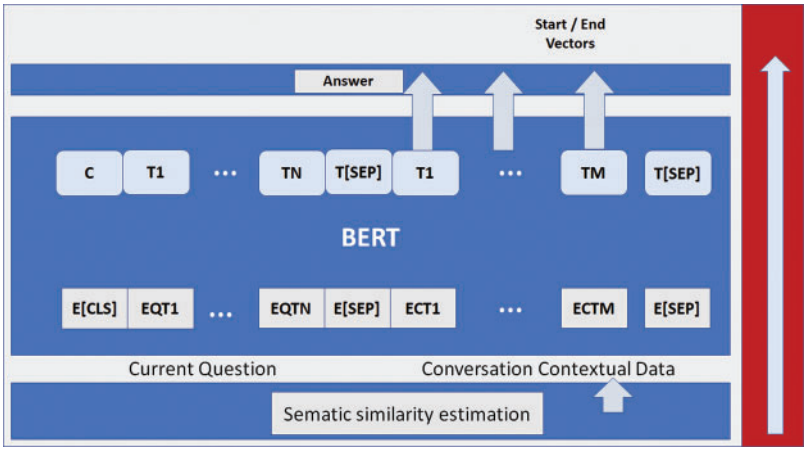

Figure 5: ConvQA using conversation context inference

Fig. 6 gives the question "Microsoft edge is not responding." The model successfully inferred relevant conversation context to the question and predicted the answer based on the inferred context. Only a single turn was needed to answer the question.

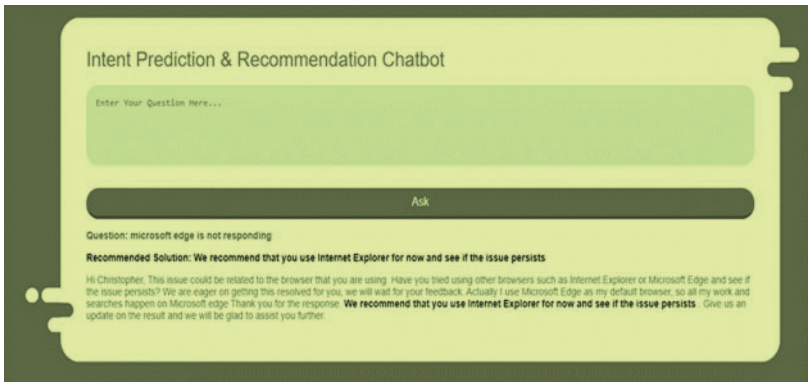

Figure 6: Output showing the ConvQA based on past conversations

4 Experiments

This system first describes the MSDialog dataset and how it applies to the research problem and then described the experimentation approach for SSE, and lastly, performed different evaluation results.

4.1 Data Set

We conduct the model experiment on ConvQA based on SSE using the MSDialog dataset. This dataset contains interactive dialogues between users seeking technical support and answer providers. The answer providers are, in this case, the Microsoft stuff and some experienced Microsoft product users. The answer (user intent) is the user's intent to get a question related to Microsoft products. The dataset consists of 35000 technical support conversational interactions, with over 2000 dialogues selected for user intent annotations. Each dialogue will comprise at least 2 to 3 turns, 2 participants and 1 correct answer. Tabs. 1 and 2 give the description and statistics of the dataset, respectively.

4.2 Simulation

The model used PyTorch as our framework for Deep Learning (https://pytorch.org), also we used the uncased pre-trained BERT model and then using the PyTorch-transformers package from hugging face (https://github.com/huggingface/pytorch-transformers), which includes different utilities and scripts for performing different NLP tasks such as ConvQA. The pre-trained BERT model comes with its vocabulary of words; therefore, extracting words from the current dataset is unnecessary. BERT consists of the uncased and cased model, and for this work, we use the uncased model, which is not case sensitive. Also, the model uses ConvQA annotation for defining the answer spans from the conversation contexts. This will split the list of conversations from the MSDialog dataset into training and validation sets. It uses different optimizers for the model to find the model with the best performance (BertAdam optimizer) and apply early stopping based on the validation set. This model applies gradient clipping using a max-norm of 1.0. The batch size for the training process is 2. For all models, the maximum length of the input text sequence

is set to 384 *tokens per sequence*, the maximum answer length is set to 512 *tokens per sequence*, the document striding is set to 128, and the maximum sequence length is set to 512 *tokens per sequence*. The learning rate of the model is set to 2×10^{-5}. It performs checkpoints on every iteration step and tests on the validation set.

Table 1: MSDialog data description and classification

Code	Label	Description	Example
OQ	Origins Question	The QA dialogue is started when a user asks the first question	Can a computer purchased with Windows 10 be downgraded to Windows 7?
RQ	Repeat Question	Posters other than the user ask a question that has already been asked	I'm having the same issue...
CQ	Clarifying Question	Users of agents request Clarification in addition to questions	Your advice is insufficiently detailed. I'm not sure what you mean when you say...
FD	Further Details	More information is available by Users/Agents	Hello, and I apologize for any inconvenience in responding. The data you require is...
FQ	Follow Up Question	Users' follow-up on significant matters by requesting follow-up questions	Thank you. I'd like to ask you a reasonable question... If I were...
IR	Information Request	Agents request information about the users	What is the model of computer's Have you tried to download it...?
PA	Potential Answer	Agents' potential response	Hello. To modify your PIN in Windows 10, follow these instructions...
PF	Potential Feedback	Users give positive feedback on result data	Hello, that was precisely the correct fix. Everything is now in order, Tx!
NF	Negative Feedback	Users focus on providing negative feedback on ineffective answers	Thanks for the update, but the preliminary steps below did not resolve the issue...
GG	Greeting/Gratitude	Users or agents may respect or show appreciation to one another	Thanks for sharing the time to answer my question...
JK	Junk	The comment contains no useful information	Emojis are emojis. Sigh... The moderator has closed the thread...
O	Others	Tweets are unable to be classified, and they use other classes	Not Applicable

Table 2: MSDialog data statistics

Item	MSDialog-Complete	MSDialog-Intent
#Dialogs	35,000	2,199
#Utterances	300,000	10,020
Avg. #Participants	3.18	2.79
Avg. #Turns Per Dialog	8.94	4.56
Avg. #Words Per Utterance	75.91	65.16

4.3 Evaluation Metrics

The evaluation of this model is based on two metrics that are, Exact Match and F1 scores. The Exact Match calculation is a binary measure. Check to see if the answer from this model exactly matches the answer from the validation set. The F1-score, on the other hand, is less strict; it computes the average overlap between the BERT model's response and the answer from the validation set. This score is taken as the proportion of the precision and recall of the answers, where precision is described as the ratio of words in the model answer that also appear in the quantitative measurements answer, and recall is defined as the ratio of words in the quantitative measurements answer that appears to be correct. For example, if the actual answer is *"You cannot use that chart type if the data is already aggregated"*, and this model predicted, *"You cannot use that chart type."* This would have high precision but lower recall, but if predicted, *"You cannot use that chart type if the data is already aggregated in Excel,"* this would have high recall but lower precision. This example also shows why the F1 scores are necessary, as answers can be presented in more than one way. Both answers will be allocated an exact match score of *'0'* if there is an overlap in the ground truth answer and the predicted answer. However, the predicted answer spans are primarily correct; hence this focused more on improving the F1-score of this model than the exact match.

4.4 Baselines

In this section, compare the evaluation metrics following previous work as a baseline model. In addition to analyzing baseline performance, analyze the performance of the proposed model over the MSDialog dataset. This model considers several models with different model parameters as baselines for conversational question answering using SSE. The methods used for comparison are described in detail as follows:

- *BERT with History Answer Embedding (HAE):* A history answers embedding for ConvQA Attentive
- *History Selection for ConvQA:* Uses a History Attention Mechanism (HAM) for ConvQA
- HAM using Bert Large
- *BERT + Context Inference:* This model implements different ConvQA with BERT, and we predict the user's intent by inferring contextual data from past CS for intent determination and answer finding.
- BERT + BertAdam
- BERT + AdamW
- BERT + FusedAdam

5 Result and Discussion

Several experiments were conducted on the MSDialog dataset for CQA based on SSE using different parameters. The experimental group compared the Exact Match and F1 scores of the proposed methods and the baseline models. Tab. 3 shows the results of the experiments.

Table 3: Each row displays validation/test scores for the respective model

Model	Optimizer	EM	F1
BERT + HAE	–	–	63.1/62.4
HAM	–	–	64.4
HAM (BERT–Large)	–	–	65.4
BERT + Context Inference	FusedAdam	25.82	82.566
BERT + Context Inference	BertAdam	25.89	83.151
BERT + Context Inference	AdamW	25.89	83.63

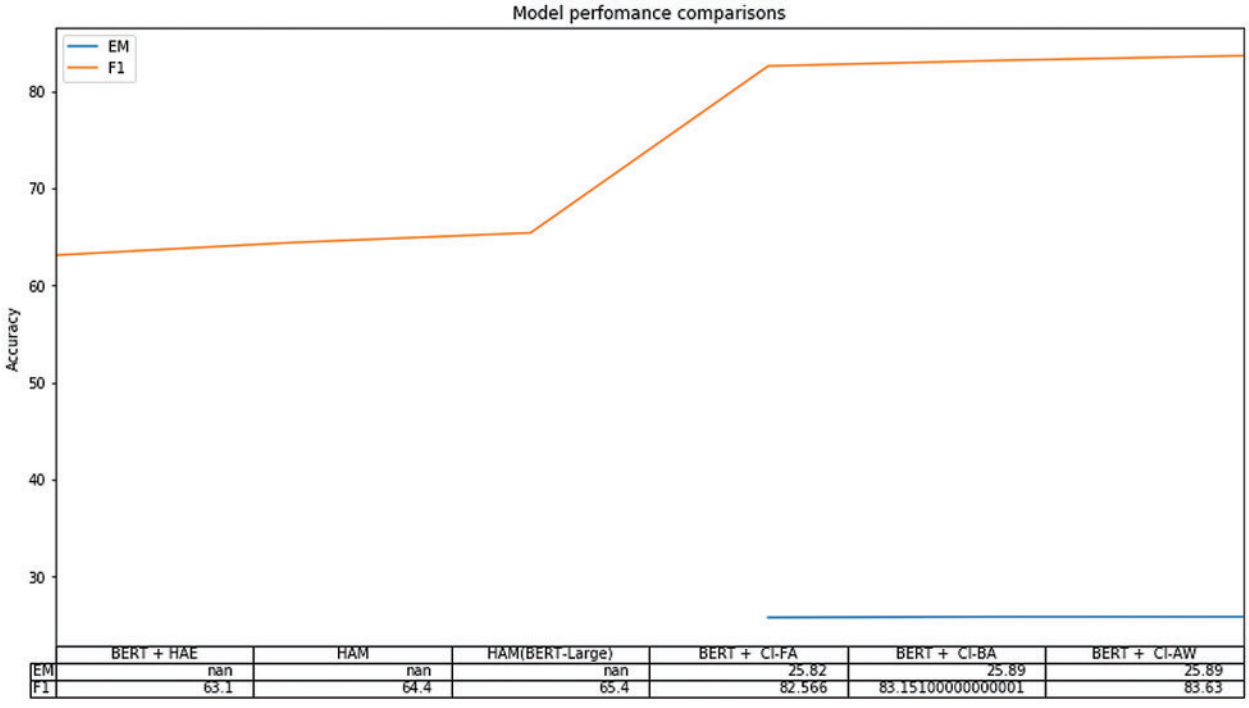

Figure 7: Each row displays validation/test scores for the respective model

The BERT model is evaluated based on two evaluation metrics: Exact Match (EM) and F1 scores are compared to the baseline models. Fig. 7 shows that HAM (BERT-Large) brings a slightly higher performance than HAM and BERT + HAE, achieving the best results among baselines. This suggests that methods of historiography attention are essential for conversation

historiography modeling and response embedding. Furthermore, our proposed model achieved much higher accuracy by using past CS than the baseline models.

Fig. 8. The proposed model, BERT + Context Inference, obtains a substantially higher performance on answer prediction than baseline methods, showing the strength of our model on that task. Also, the performance of our model is affected slightly by using different optimizers, as shown in Figs. 9 and 10 for both F1 and EM scores, respectively. However, the model sees no significant differences when using different optimization functions. Increasing the sequence length of the query and the maximum input sequence significantly improves the F1 scores, suggesting that a model that can take more than 512 tokens as input can achieve even better results. Also, train the model using different model parameters. Experimental results from our models as well as the other *state-of-the-art* model are shown in Tab. 3, where the first model uses the BERT-HAE, the second model uses the HAM, the third model uses the HAM on BERT Large, and the rest of the models represent the proposed BERT model with context inference implemented across different model parameters.

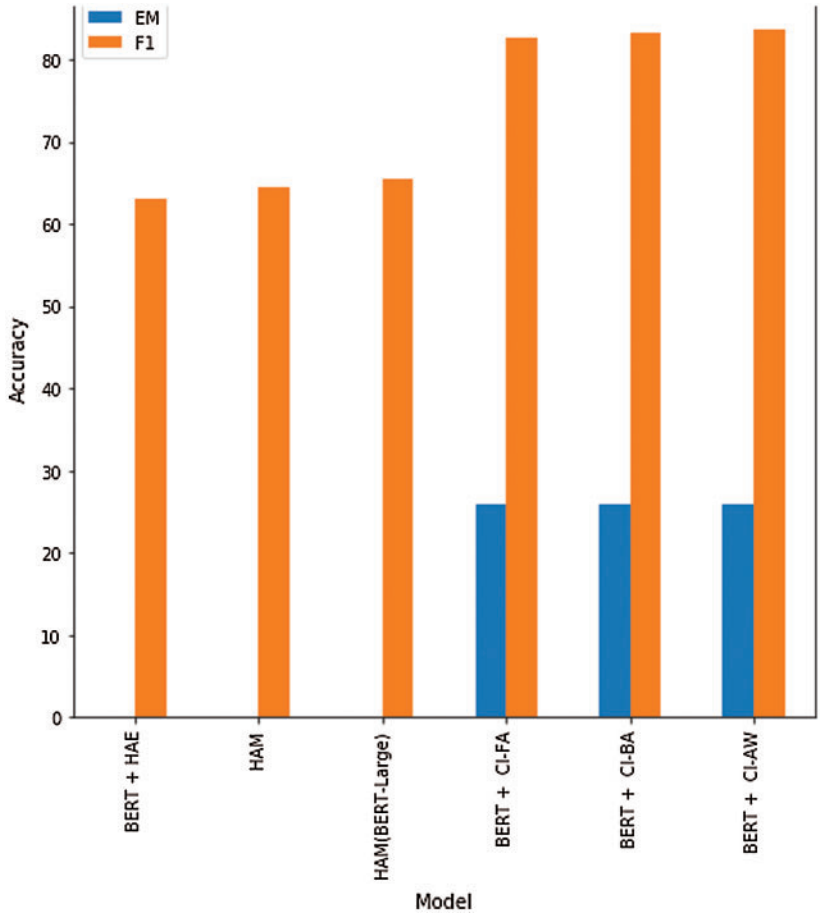

Figure 8: Each bar displays F1/EM-scores for the respective model

Tab. 3, the new proposed model using AdamW optimization achieved the F1-score of 83.63 after 10 epochs, using default parameters, and it performed better than BERT models, which have

the same settings except for the AdamW optimization. In particular, on the MSDialog dataset, the model using AdamW Optimization improves by 0.07% EM and 1.06% F1 compared to the Fused Adam models, and 0.0% EM and 0.479% F1 over the BertAdam models. Leveraging the BERT-Large model makes multi-passage BERT even better on MSDialog datasets.

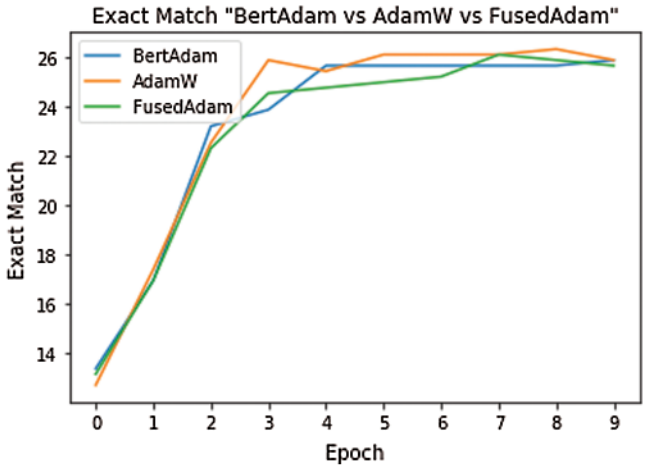

Figure 9: Exact match scores based on different optimizers at different epoch thresholds

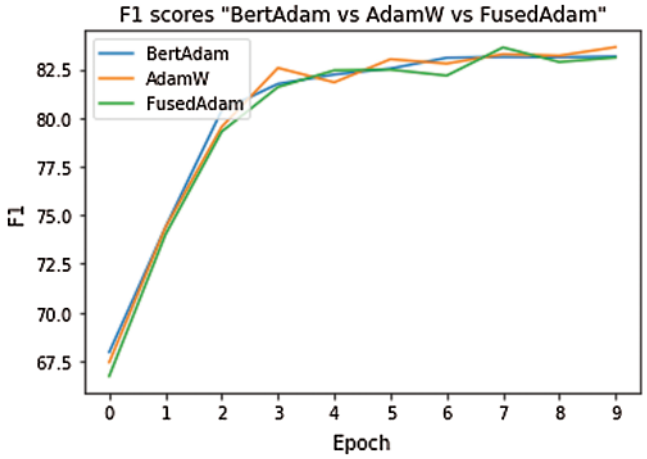

Figure 10: Accuracy of the model using different optimizers at different epoch thresholds

Fig. 11 shows the results with different optimization functions. In all cases, inferring past conversation contextual data was practical. However, using different optimizers had little effect on accuracy. The results show that the best accuracy was obtained in the case of using the AdamW optimizer. This model uses past conversation contextual representations trained over the question-and-answer pairs from previous CS. A new method captured the user's informational needs based on interaction from past CS, and it achieved reasonably high accuracy. Pre-trained BERT was trained with two input segments, which makes it suitable for this task. However, providing accurate answers is highly dependent on the richness of the researcher's knowledge base

in terms of contextual data availability. Past contextual data on the knowledge base (MSDialog) can include the context unrelated to the current question. Since this model infers contextual data based on the current question and past contextual data similarities, the context with the highest score may not be related to the given question and can decrease the accuracy. Notably, it is essential to update the knowledge base with new context from CS. Future work will constitute context selection only related to the current question and using context from other knowledge bases for better performances.

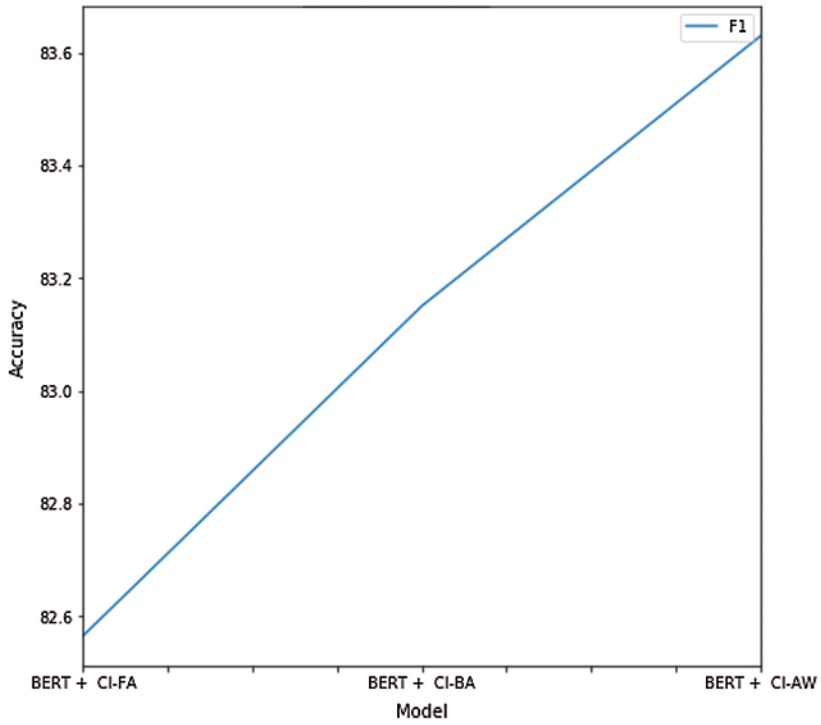

Figure 11: Comparison of the proposed model using different optimizers

6 Conclusion and Future Work

This paper introduced SSE and context inference, a method for determining user intent and providing answers from partial/limited information from users based on past CS without engaging in multi-turn interactions with the system. This system can understand limited information from user questions and can infer relevant contextual data from past conversations and at the same time provide exact answer spans using questions of the same domain area. Modelling utterances as conversational context using BERT deliver a significant improvement in ConvQA over the existing model. Using BERT for contextual representations enhanced the performance of this model. When it comes to multi-turn interactions in CS, available datasets have no precise contextual information for modeling good models; however, the MSDialog dataset provided the necessary multi-turn information-seeking conversations for proposed work. This work will create more research interest in intent prediction and CSS, critical for simulating human-human interactions in CA.

In the future, researchers design to combine our history design methodology with a learned history analysis algorithm for ConvQA.

Funding Statement: The authors received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts to report regarding the present study.

References

[1] K. Chen and A. Salman, "Learning speaker-specific characteristics with a deep neural architecture," *IEEE Transactions on Neural Networks*, vol. 22, no. 11, pp. 1744–1756, 2011.

[2] E. Ntoutsi, P. Fafalios, U. Gadiraju, V. Iosifidis, W. Nejdl *et al.,* "Bias in data-driven artificial intelligence systems—An introductory survey," *WIREs Data Mining and Knowledge Discovery*, vol. 10, no. 3, pp. 60, 2020.

[3] E. T. Bradlow, M. Gangwar, P. Kopalle and S. Voleti, "The role of big data and predictive analytics in retailing," *Journal of Retailing*, vol. 93, no. 1, pp. 79–95, 2017.

[4] F. Richardson, D. Reynolds and N. Dehak, "Deep neural network approaches to speaker and language recognition," *IEEE Signal Processing Letters*, vol. 22, no. 10, pp. 1671–1675, 2015.

[5] J. Liu, Y. Jiang, Z. Li, X. Zhang and H. Lu, "Domain-sensitive recommendation with user-item subgroup analysis," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 4, pp. 939–950, 2016.

[6] J. R. Saura, "Using data sciences in digital marketing: Framework, methods, and performance metrics," *Journal of Innovation and Knowledge*, vol. 6, no. 1, pp. 92–102, 2020.

[7] L. Wang, A. Sy, L. Liu and C. Piech, "Deep knowledge tracing on programming exercises," in *ACM Proc. of the Fourth ACM Conf. on Learning @ Scale, (L@S '17)*, New York, NY, USA, pp. 201–204, 2017.

[8] M. Kim, S. Lee and J. Kim, "A wide and deep learning sharing input data for regression analysis," in *IEEE Int. Conf. on Big Data and Smart Computing (BigComp)*, Busan, Korea (South), pp. 8–12, 2020.

[9] M. Vacher, S. Caffiau, F. Portet, B. Meillon, C. Roux *et al.,* "Evaluation of a context-aware voice interface for ambient assisted living: Qualitative user study vs. quantitative system evaluation," *ACM Transactions on Accessible Computing*, vol. 7, no. 2, pp. 1–36, 2015.

[10] M. Zubani, L. Sigalini, I. Serina and A. E. Gerevini, "Evaluating different natural language understanding services in a real business case for the Italian language," *Procedia Computer Science*, vol. 176, pp. 995–1004, 2020.

[11] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, pp. 788–798, 2011.

[12] A. Baba, S. Yoshizawa, M. Yamada, A. Lee and K. Shikano, "Acoustic models of the elderly for large-vocabulary continuous speech recognition," *Electronics and Communications in Japan, Part 2*, vol. 87, no. 7, pp. 49–57, 2004.

[13] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado and J. Dean, "Distributed representations of words and phrases and their compositionality," *Advances in Neural Information Processing Systems*, vol. 2, pp. 3111–3119, 2013.

[14] Y. Li, L. Lu and X. Li, "'A hybrid collaborative filtering method for multiple-interests and multiple-content recommendation in e-commerce," *Expert Systems with Applications*, vol. 28, no. 1, pp. 67–77, 2005.

[15] A. Shafik, A. Sedik, B. Abd El-Rahiem, E. S. M. El-Rabaie, G. M. El Banby *et al.,* "Speaker identification based on radon transform and CNNs in the presence of different types of interference for robotic applications," *Applied Acoustic*, vol. 177, no. 107665, pp. 1–12, 2021.

[16] M. A. Khanand and Y. Kim, "Deep learning-based hybrid intelligent intrusion detection system," *Computers, Materials and Continua*, vol. 68, no. 1, pp. 671–687, 2021.