

Received October 28, 2021, accepted November 11, 2021, date of publication November 16, 2021, date of current version November 24, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3128616

Neuro-Symbolic Speech Understanding in Aircraft Maintenance Metaverse

AZIZ SIYAEV¹ AND GEUN-SIK JO^{1,2}, (Senior Member, IEEE)

¹Artificial Intelligence Laboratory, Department of Electrical and Computer Engineering, Inha University, Incheon 22212, South Korea

²Augmented Knowledge Corporation, Incheon 22212, South Korea

Corresponding author: Geun-Sik Jo (gsjo@inha.ac.kr)


This work was supported in part by the BK21 Four Program (Pioneer Program in Next Generation Artificial Intelligence for Industrial Convergence) by the Ministry of Education (MOE), South Korea, under Grant 5199991014250; in part by the National Research Foundation of Korea (NRF); in part by the Ministry of Science and ICT (MSIT), South Korea, through the Information Technology Research Center (ITRC) Support Program supervised by the Institute for Information and Communications Technology Promotion (IITP) under Grant IITP-2017-0-01642; and in part by the INHA UNIVERSITY Research Grant.

ABSTRACT In the emerging world of metaverses, it is essential for speech communication systems to be aware of context to interact with virtual assets in the 3D world. This paper proposes the metaverse for aircraft maintenance training and education of Boeing-737, supplied with legacy manuals, 3D models, 3D simulators, and aircraft maintenance knowledge. Furthermore, to navigate and control operational flow in the metaverse, which is strictly followed by maintenance manuals, the context-aware speech understanding module Neuro-Symbolic Speech Executor (NSSE) is presented. Unlike conventional speech recognition methods, NSSE applies Neuro-Symbolic AI, which combines neural networks and traditional symbolic reasoning, to understand users' requests and reply based on context and aircraft-specific knowledge. NSSE is developed with an industrially flexible approach by applying only synthetic data for training. Nevertheless, the evaluation process performed with various automatic speech recognition metrics on real users' data showed sustainable results with an average accuracy of 94.7%, Word Error Rate (WER) of 7.5%, and the generalization ability to handle speech requests of users with the non-native pronunciation. The proposed Aircraft Maintenance Metaverse is a cheap and scalable solution for aviation colleges since it replaces expensive physical aircraft with virtual one that can be easily modified and updated. Moreover, the Neuro-Symbolic Speech Executor, playing the role of field expert, provides technical guidance and all the resources to facilitate effective training and education of aircraft maintenance.

INDEX TERMS Aircraft maintenance education, Boeing-737, deep learning, industry 4.0, metaverse, mixed reality, neuro-symbolic AI, transformer, smart glasses, speech recognition.

I. INTRODUCTION

Connecting in through digital spaces is becoming an essential part of everyday life, and the concept of metaverses creates environments where people can meet in virtual spaces and socialize. Being a mirror of the real world with enhanced features, metaverse provides assets and values in digital form various industries can benefit from. Since recently, the world is hit by pandemics, industries seriously consider migrating their workflow, training, and education into online alternatives - metaverses to enable employees to work remotely and businesses to continue to function [1], [2]. Marketing, Economy, Culture, Entertainment, Education all will be inevitably

The associate editor coordinating the review of this manuscript and approving it for publication was Andrea F. Abate .

integrated into metaverses, connecting people and bringing their values in a virtual, yet realistic world [3].

In this work, we proposed the Aircraft Maintenance Metaverse of Boeing aircraft [4], where professionals and trainees can meet in the collaborative space of maintenance field equipped with aircraft-specific virtual assets of Boeing-737. Being fully virtual and replacing expensive hundreds of millions of dollars physical airplanes [5], the proposed metaverse is a cost-effective solution for institutions training on outdated aircraft models. The proposed Aircraft Maintenance Metaverse is an education-oriented virtual mixed reality space that consists of all supportive materials for maintenance operations, and trainees wearing smart glasses HoloLens 2 [6], can meet together to simulate aircraft maintenance. What is more, legacy manuals such as Aircraft

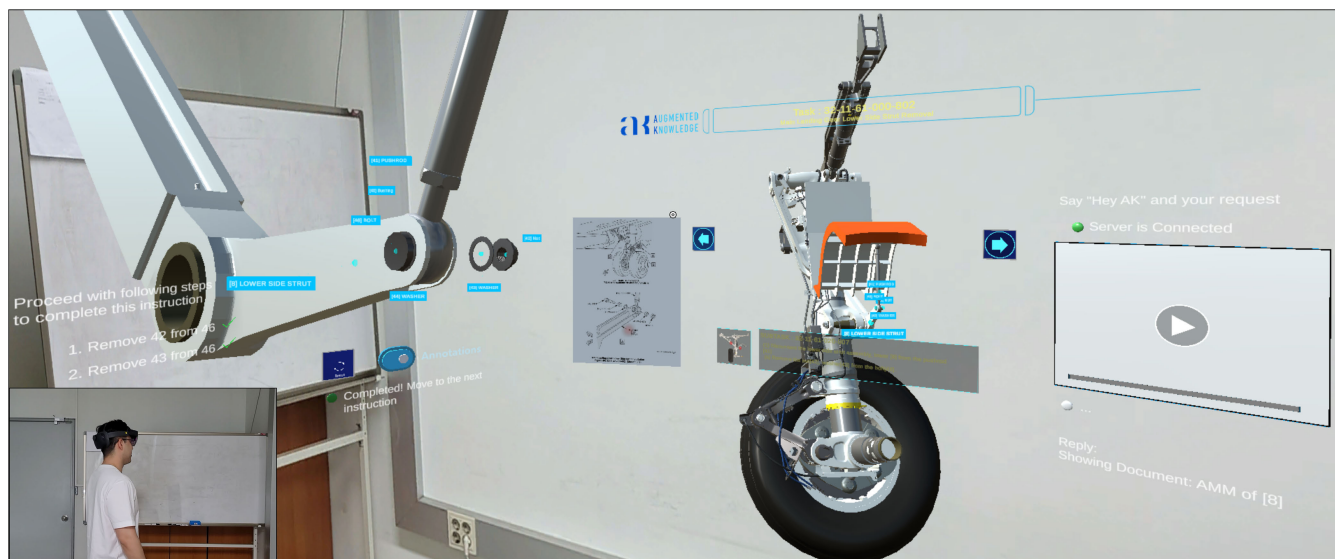


FIGURE 1. Proposed aircraft maintenance metaverse. The first-person view snapshot is taken with HoloLens 2 by the user shown left below.

Maintenance Manuals (AMM) [7] and Illustrated Parts Catalogue (IPC) [8], being used until now, are enhanced with 3D manuals, which work as a 3D simulator for practicing various operations on 3D models as if on physical parts of the airplane.

The whole control in the system is driven by speech, which creates a feeling of conversation with the 3D world. To effectively control virtual assets and environment in the metaverse, we proposed the Neuro-Symbolic Speech Executor (NSSE) which is based on Neuro-Symbolic AI that combines superior abilities of neural networks in pattern recognition and symbolic reasoning of traditional AI. To process complex speech requests of users and reason based on context, NSSE applies several stages of neural networks and symbolic programs execution. First, a speech signal from HoloLens 2 is processed using the algorithm for Dynamic Audio Length Recoding, next, with the help of the Speech-To-Text network we convert audio data to the text, and the third component Text-To-Programs converts the text to the functional programs. Lastly, the Symbolic Programs Executor, which maintains the manual knowledge and contextual information, executes proposed programs to generate the result and provide audio and visual feedback to a user. Neural networks Speech-To-Text and Text-To-Programs in the NSSE are built from Transformers architecture, which perfectly suits the language modeling tasks, and in comparison with other works [9]–[11] that applied RNN, Transformer architecture has significant advantages.

To train neural components of NSSE, we applied an industrially cheap and efficient approach: for training data collection, no people are involved, instead, Text-To-Speech API services synthesized data for training; existing language Speech-To-Text models are fine-tuned with aircraft specific domain vocabulary. We evaluated the performance of NSSE on several metrics using test sets containing both native and non-native pronunciation of speech requests from real peo-

ple. The experiments demonstrated that existing pre-trained Automatic Speech Recognition models trained on general datasets do not handle samples with professional words well, thus, need to be fine-tuned with additional domain-specific data, and as result, fine-tuned models show 50% better performance. In addition, considering all test data, NSSE demonstrated an average Word Error Rate of 7.5% and overall accuracy of 94.7%, which is considered to be a good quality for automatic speech recognition tasks to be used in the industry [12].

The proposed Aircraft Maintenance Metaverse with all its innovative distinctive features and virtual content creates enormous value in the field of aircraft maintenance and pushes forward the development and integration of metaverses industrially and globally. Furthermore, communication within the metaverse that happens using Neuro-Symbolic Speech Executor, makes it possible to talk to the virtual world and get the innovative technological experience for training and education.

In the following sections, we discuss background information about combining Extended Reality and speech understanding methods, Neuro-Symbolic AI, and Transformers architecture. Next, we presented Aircraft Maintenance Metaverse and its features. In section 4, the proposed Neuro-Symbolic Speech Executor is explained in detail, which is assessed later in the Evaluation section. To summarize the contributions and results of this paper, the Conclusion section is provided.

II. BACKGROUND

A. COMBINING EXTENDED REALITY AND SPEECH RECOGNITION

Extended Reality (XR) as a technology captures all real-and-virtual merged environments including Virtual Reality (VR), Augmented Reality (AR), Augmented Virtuality (AV), Mixed

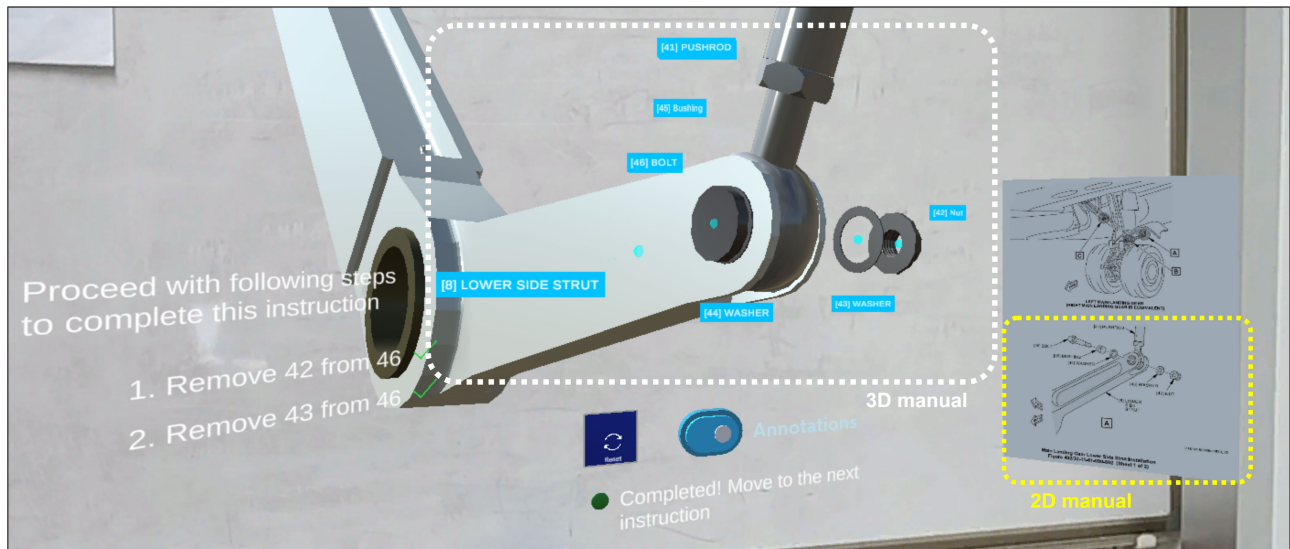


FIGURE 2. Aircraft maintenance manual 3D simulator.

Realities (MR), and combining it with Speech Recognition makes a union that brings enormous value to various aspects of people's lives and industries. This includes helping people with disabilities, improving the education process, or easing a workflow in the industry.

References [13]–[16] discuss the ways to solve the communication problems among D(deaf, hearing impaired, or hard of hearing people using AR and Speech Recognition technologies. Making a narrator's speech illustrated to deaf people on display by creating real-time Augmented Reality "live subtitles" while hearing a talk, helps them to hear and feel the environments in a visual way, further overcoming the communication barrier between deaf and people, who do not know sign language.

Education also benefits from learning through a combination of AR and speech recognition technologies. One example is learning new languages [17], where Augmented Reality offers an enhanced environment that influences non-native "children's experience and knowledge gain during the language learning process" [17]. AR together with Speech Recognition facilitates enjoyment during learning, enables young children to interact with virtual objects to cope with certain tasks such as learning words for basic colors, 3D shapes, and spatial objects relationships faster and easier [17].

Various operations at industrial workflow can be automated or enhanced with XR and Speech recognition: XR helps to simulate or digitalize the working process, speech commands help to control operations, saving time and being flexible, efficient, and economical form of communication. Reference [18] presents the concept of implementation of AR and speech interface for controlling lifting devices, eliminating the need to be physically present at the site for crane works. Another example in [19] shows how MR aircraft maintenance can be facilitated with speech commands, where the digital twin of aircraft is used instead of real one. Next, we discuss certain ways to embed speech understanding technologies in XR applications.

B. DEVELOPMENT OF SPEECH UNDERSTANDING IN EXTENDED REALITY

To develop and embed Speech recognition in XR, it is crucial to understand the nature of user requests, functionalities the speech communication addresses, and environments an application is built for. For example, speech requests that consist of only predefined, static, and short-sentenced commands such as "Play", "Stop", "Next Image" can be easily handled by offline built-in voice control in mobile devices such as smart glasses HoloLens [20].

On the other hand, for users' requests that have longer sentences, flexible semantic structures, and referring to the same functionality, classification neural networks can be applied. For example, commands "Show me the next object", "Display the next object", "Move to the next item" map to the action, which displays the next object in order, thus, the classification model can map voice signal features into one action class out of a set of predefined categories. Usually, CNN-based neural model architectures are utilized for audio classification due to superior abilities to extract data features. Depending on the type of audio features, 1D or 2D convolution filters are used. In the case of processing raw audio signals, 1D convolution is applied [21]–[23]; for MFCC or Log spectrum features - 2D is used [24], [25]. Similarly, our preceding work [19] concentrated on the implementation of speech commands with the help of a custom bilingual CNN neural network that extracts MFCC features from spoken audio data in English and Korean and converts it to one of 8 classes that triggers a certain action to be taken. For example, "Please, play tutorial video" triggers a media player to start a reference video. The network takes audio MFCC features and produces 2 results: action class and identified language. Here, speech communication calls the operational functions of the application.

There are cases when an XR system requires transcription of speech, and conversion of audio signals to text is achieved with the help of Automatic Speech Recognition (ASR)

techniques such as [26]–[31], which built acoustic models to map signal waves to sequences. Reference [27] applied a fully convolutional model that takes raw audio as input and computes speech representation, whereas, [26] utilized Recurrent Neural Network (RNN). Reference [28] combined the Connectionist Temporal Classification (CTC) [32] predictions from the attention-based decoder and LSTM-based language model to obtain results. In the XR system, such networks are used online, not in the actual mobile device, since they require space and computing power in order to be utilized. Nevertheless, inference of neural models, either classification or ASR networks, does not depend on contextual information. However, in this paper, we addressed the case and demand when context matters.

In this work, we take into account aircraft maintenance manuals, which are the legal documents that have to be strictly followed by mechanics, since the consequences of operational mistakes during MRO can be devastating and lethal. Therefore, having operational control with speech communication requires a strong relationship with manuals that represent contextual information. Manuals have knowledge and hierarchy in tasks, subtasks, instructions, aircraft parts, 2D manuals, 3D objects, tools, warnings, cautions, and etc. All items in the document are linked, creating knowledge graphs that must be referred to while the maintenance process. Therefore, the development of speech communication and control with simple-structured deep learning networks cannot handle all resources and relations in manuals and consider context while inferencing. Overall, the speech interaction system needs a logic-based part that reasons based on contextual information and compliments pattern recognition abilities of neural networks. Thankfully, recent advancements in the field of neural networks - Neuro-Symbolic AI combines the abilities of both neural networks and symbolic AI for logic-based reasoning.

C. NEURO-SYMBOLIC AI

The Neuro-Symbolic AI - a new methodology for AI, to enhance the strengths of statistical AI, such as machine learning, with the complementary capabilities of symbolic or classical AI, which is based on knowledge and reasoning [33]. “The term neural in this case refers to the use of artificial neural networks, or connectionist systems, in the widest sense. The term symbolic refers to AI approaches that are based on explicit symbol manipulation” [34]. Neural and symbolic AI approaches differ in the representation of information within an AI system. For symbolic systems, the representations are explicit, manipulated by symbolic means, and understandable by humans. However, in neural systems, representations are usually by means of weighted connections between neurons [34]. The main goals of Neuro-Symbolic AI are to solve complex problems with the ability to learn on a small amount of data, providing to users understandable reasons on each decision and controllable actions, which is crucial when integrating AI in the industry [33].

The rise of Neuro-Symbolic AI started with several works that unleashed opportunities of this approach. Reference [9]–[11], [35] proposed techniques based on neural-symbolic AI for visual and language understanding to perform joint learning of concepts from images and related question-answer pairs. Applying deep learning for visual recognition and language understanding, and traditional AI in symbolic program execution for reasoning, their approaches are able to answer various relational and conceptual questions from a given image. Reference [9]–[11] used the CLEVR dataset [35] for Visual Question Answering (VQA) systems to reason and answer questions about visual data. Images in the dataset consist of simple 3D shapes such as cylinder, cube, sphere. Each object has its own color (red, green, blue, etc.), material (rubber or metal), and size (small or large), and is located in a certain relational position to other objects in a scene (left, right, behind, and in front of a particular object). In order to reason in these scenes, researchers in [35] introduced functional programs for each question in CLEVR, where a program can be executed on a scene graph, giving the answer to the question from the image. The proposed programs contain querying, counting, or comparing operations that in combination give a particular result.

Having closer look at [9], the proposed approach separates vision and language understanding from reasoning. First, using neural networks an image scene is parsed, and the question is being understood by converting it to functional programs. The parsed image information is structured in knowledge. Next, the reasoning applies symbolic execution of programs based on the knowledge to give the answer to the question. Reference [9] applied Mask R-CNN and CNN networks to extract structural scene representation. In order to process questions and generate programs, a sequence to sequence model with an encoder-decoder bidirectional LSTM [36] encoder is applied. Having various advantages such as robustness to complex programs, small training data, the method achieved excellent accuracy on the CLEVR dataset.

“The integration of learning and reasoning is one of the key challenges in artificial intelligence and machine learning today” [37] and there are many questions that remain open: semantics of neural-symbolic approaches, explainability, potential applications, being able to generalize to tasks with minimal or no domain-specific training, etc. The research community slowly realizes the inherent limitations of pure deep learning approaches, and additional background knowledge with logical reasoning is a natural path to attempt to further improve deep learning systems [34]. In this work, we incorporated the work of neural networks based on an architecture called Transformers, therefore, the next section introduced this decent technology.

D. TRANSFORMERS

The Transformer is a novel neural network architecture proposed in [38] based on attention mechanisms, which entirely dispensing with recurrence and convolutions. Before the

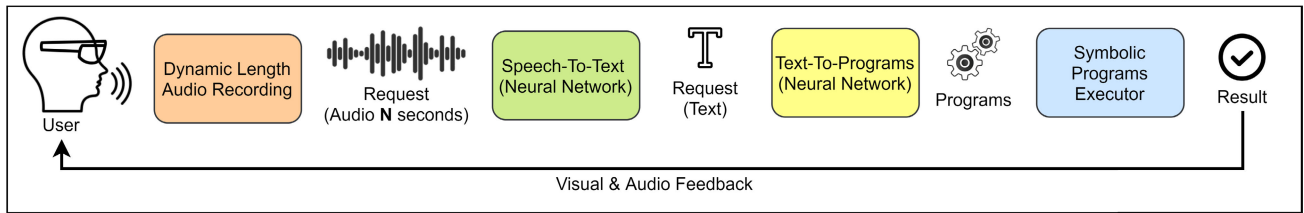


FIGURE 3. Action flow of neuro-symbolic speech executor.

introduction of Transformer, RNNs were considered to be de facto standard for approaches to language understanding tasks such as language modeling, machine translation, and question answering [39]. However, the work “Attention Is All You Need” [38] presented the architecture that outperforms both RNN and CNN models on various translation benchmarks, where the main idea is relying entirely on an attention mechanism and at the same time eschewing recurrence in order to find correlations between input and output [38].

Various works [38]–[40] demonstrated the superiority of Transformers over RNNs in natural language processing tasks. Recurrent neural networks work by treating natural language as a time series, where every word modifies the meaning of all the words that came before it. RNN looks at one word at a time and creates a representation to further contextualize that representation with the representation of the next word [41]. Comparing the Transformer and RNN architectures, Transformer learns sequential information via a self-attention mechanism processing a sentence as a whole, whereas RNNs extract representations word by word, which does not allow parallel processing, therefore, training process in Transformer is more efficient since can be distributed over multiple GPUs. What is more, Transformers do not rely on past states to capture dependencies with previous words but process a sentence as a whole, and multi-head attention and positional embeddings provide information about the relationship between different words, however, RNN architecture keeps learned information through past states, where each state is assumed to be dependent only on the previous state, thus creating issues in long dependencies. Therefore, Transformer can take a “word or even pieces of words and aggregate information from surrounding words to determine the meaning of a given bit of language in context” [39].

Taking into account all advantages of the given approach, we build our language understanding models in this work, such as speech recognition and translation, based on Transformers architecture, and achieved successful results. In the next sections, the contributions of this work are presented.

III. AIRCRAFT MAINTENANCE METAVERSE

A. OVERVIEW

Aircraft maintenance metaverse is a collaborative space that lets people in the field of Maintenance Repair Operations (MRO) get together to operate on aircraft-specific virtual assets. The Metaverse as a term is used to describe the concept of a future iteration of the internet, made up of persistent, shared, 3D virtual spaces linked into a perceived virtual

universe [42]. At the same time, the metaverse we created is a learning place for trainees to operate on virtual aircraft that has supportive materials and functions that facilitate the maintenance training. Being guided by the virtual manuals and having all you need to get the job done, the metaverse creates an effective workflow of training. Considering the fact that the contemporary world is hit by COVID-19, and various industries migrate from traditional work or formal education to online alternatives enhancing Society 5.0 [43], virtual spaces such as the proposed metaverse create potential solutions to deal with the challenges presented by the pandemic. The metaverse creates interoperable gateways to connect worlds, functioning as an all-encompassing, unified portal and hub [1]. Same way, we combined the real world with the world of virtual aircraft and maintenance.

Considering the cost of physical airplanes, which may reach more than hundreds of millions of dollars (Boeing-737 costs 100 million dollars [5]), the proposed aircraft maintenance metaverse represents a potential solution for various aviation colleges and schools that arrange training on outdated aircraft models. Virtual models of aircraft in the metaverse can be easily updated or replaced. In addition, while working on a physical part (i.e. aircraft landing gear), usually special equipment is required just to transport or install it, since the weight of such components is huge. In contrast, various interaction mechanisms in smart glasses let users manipulate assets in an intuitive way with just a touch of fingers. Hence, the role of the metaverse in the industry is enormous, due to saving a vast amount of resources.

To access the metaverse, smart glasses HoloLens 2 [6] are being used: they help to project Mixed Reality into the real world and deliver an immersive experience of the 3D world. Fig. 1 illustrates the snapshot of the proposed aircraft maintenance metaverse, which is captured from a first-person view, of the user shown left below in the figure.

Exploring the figure, it can be seen that various visual components exist. First, the main asset is a particular aircraft part to work on. It is located in the center and represents a digital twin of an actual physical model. In Fig. 1 main landing gear of Boeing-737 is illustrated. The model has annotations to its component parts to let the novice users have visual clues. Second, on the right to the model, a media player, which demonstrates a video reference is placed. A tutorial video summarizes the job of fellow engineers on a particular task, helping trainees to understand the procedure to be completed. Next, to the left of the digital twin, the manual section is demonstrated. The proposed system kept the legacy

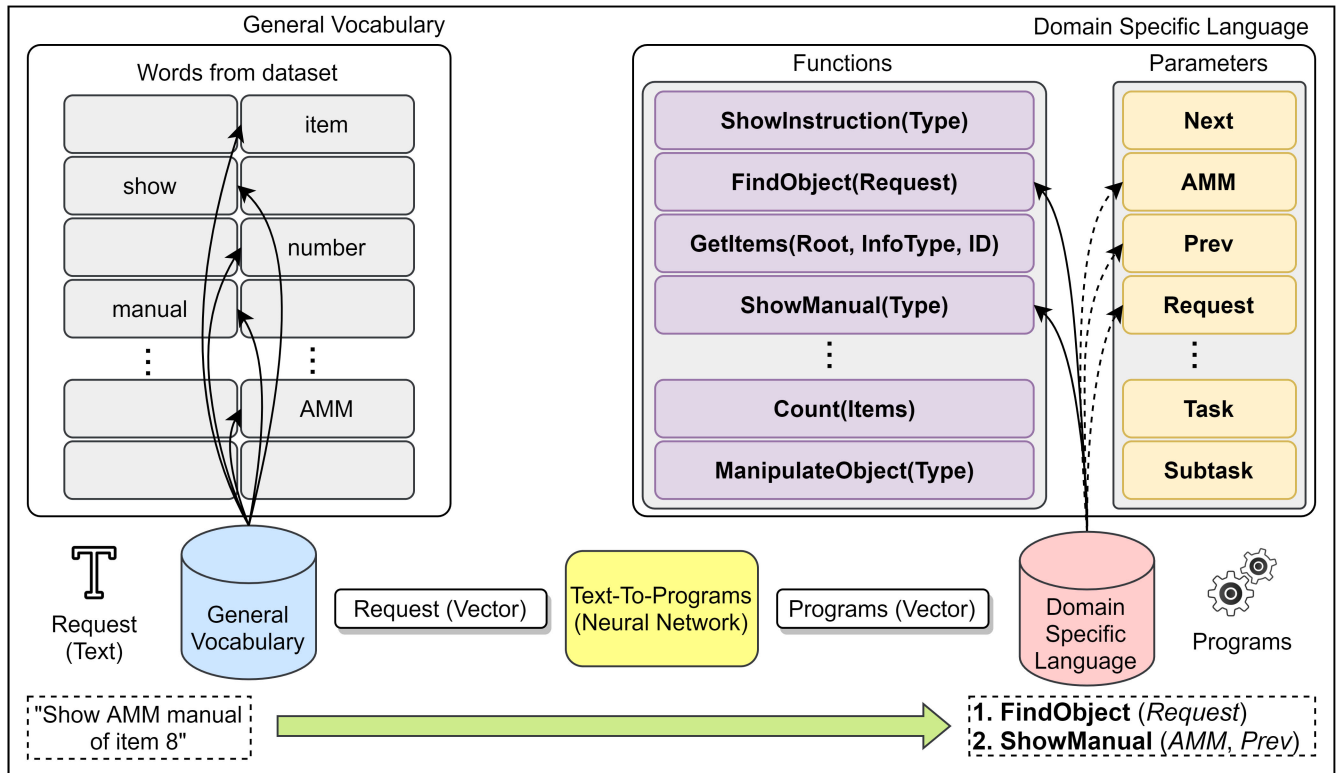


FIGURE 4. Text-to-programs action flow.

2D Aircraft Maintenance Manual [7] and introduced an innovative 3D simulator of it (see next section).

All implemented procedures in the system are based on official Boeing-737 manuals and documentation, since keeping the defined protocol is crucial for safety and effectiveness. Thus, the first step we performed in this project is converting legacy documents into a structured format to use in the system. The JSON format is used to encapsulate a massive amount of data and convert it into knowledge, and at the same time to enhance the concept of the web in the metaverse, which includes the sum of all virtual worlds and the Internet [1], [42], [44]. The system kept trusted traditional way of aircraft maintenance, at the same time, enhancing procedures with new dimensions of information, such as Mixed Reality animations, media content, and 3D manuals that innovates maintenance training and education.

B. 3D MANUAL

3D Manual represents a new way to look at traditional 2D manuals. Usually, 2D legacy manuals have figures that illustrate a certain process with annotations. They are static and show the end result as a snapshot. In Fig. 2, the example of the 2D AMM manual is presented: it shows the lower side strut removal of the main landing gear, where the 2D manual works as a reference on how to perform a particular task during the maintenance process.

The 3D manual is a model that helps to visualize a scene and individual components separately from different angles to better understand the information it references to. Introduc-

ing new dimensions to look at legacy manuals, we proposed the 3D manual that completes 2D legacy figures. In Fig. 2 3D manual is demonstrated as an addition to its 2D manual, so that trainee who refers to the figure in the book, can explore it in an understandable view.

Besides having an explorable third dimension, the 3D manual has various functionality. 2D figure encapsulates information of the task, subtask, or instruction and displays desirable end result, eliminating the process, however, with the 3D manual intermediate processes can be explored as well. In Fig. 2, 2D represents the end result of subtask execution, which has the following three instructions:

- “Remove the nut 42, washer 43 from the bolt 46”
- “Remove the bolt 46 to disconnect the lower side strut assembly”
- “Isolate the pushrod 41 from the lower side strut assembly”

On contrary, using the proposed 3D manual we can have a deep view at the instruction level and even perform it step-by-step, i.e. “Remove the nut 42, washer 43 from the bolt 46” in Fig. 2, is divided into two steps: removing 42 from 46, and 43 from 46, thus a trainee can execute a particular instruction as one, or divide it into substeps as shown. When the step-by-step execution of an instruction is performed, the visual clue as “completed” icon is shown, to have better navigation of the process (see Fig. 2).

Each subtask or instruction have their own 3D manual which can be considered as a simulator to experiment on.

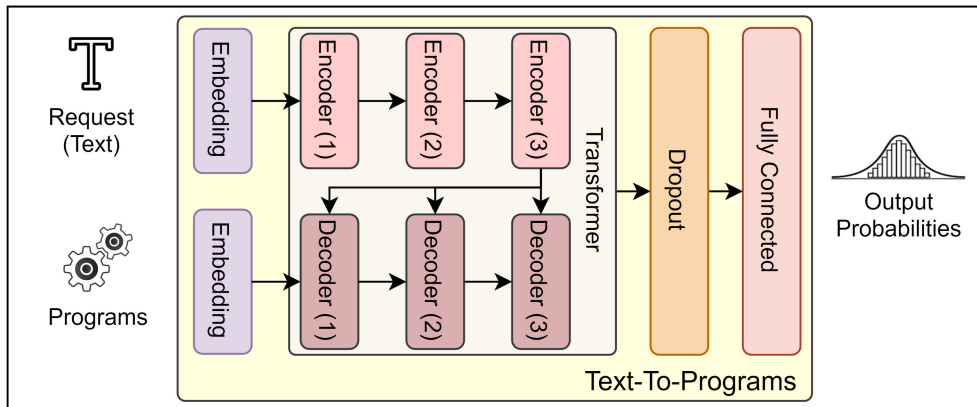


FIGURE 5. Text-to-programs network architecture.

In order to control this complex process, speech commands are used, which are handled using the proposed novel method described in the next section.

IV. NEURO-SYMBOLIC SPEECH EXECUTOR

A. OVERVIEW

Neuro-Symbolic Speech Executor is a module that incorporates the work of neural networks and symbolic reasoning to process speech requests in the proposed Aircraft Maintenance Metaverse. Combining superior abilities of deep learning in pattern recognition and traditional AI for reasoning, NSSE understands complex users' spoken commands with various semantic structures that include aircraft-specific domain vocabulary and various references to legacy maintenance manuals (i.e. "Display AMM document of item 8" - NSSE recognizes that aircraft-specific manual AMM should be demonstrated to a user and an item under number 8 in the document must be highlighted to navigate the person).

Neuro-Symbolic Speech Executor has four steps to perform inference (see Fig. 3). Firstly, a user wearing smart glasses triggers the NSSE to record his/her audio request. To do so, the keyphrase "Hey, AK!" is used. NSSE detects the trigger phrase and invokes the Dynamic Length Audio Recording (DLAR) algorithm to handle the creation of audio data from a stream of speech signals outgoing from the microphone. Depending on the length of the speech request, the output of DLAR is an n-second duration audio request. Next, the audio request is passed to the Speech-To-Text, which is a neural network for automatic speech recognition that converts raw audio data into text, extracting the transcript of the request. Thirdly, the Text-To-Programs sequence-to-sequence network takes the transcript of the speech request in the English language and matches it to the sequence of executable programs of the created domain-specific language that consists of functions and parameters. Lastly, the created programs (the combination of particular functions and parameters) are being executed one by one using the Symbolic Programs Executor to get the result and send visual and audio feedback back to a user to notify about processing and completion of the requested operations.

B. DYNAMIC LENGTH AUDIO RECORDING

Audio commands in the system may have various lengths, i.e. command "Next instruction" is 1.37 seconds and "Remove objects 42, 43 from 46" is 3.94 seconds, it is inefficient to build an audio recorder that listens only a specified amount of time. Analyzing possible speech requests that can be made (45,244 requests), on average they have 2.76 seconds length with a standard deviation of 0.87 seconds, where the shortest one is 0.54 seconds and the longest speech request is 7.61 seconds. In the case of the static audio recorder approach, listening time should be set as the maximum time in order to handle all requests. Thus, the total time for 45,244 speech commands would be 344,306.84 seconds (45,244 x 7.61), with the average time to be wasted is 4.85 seconds, because even though a speaker finished request, static recording continues to listen until the defined time.

In order to improve response time from the system, we proposed Dynamic Length Audio Recording which is a dynamic algorithm that helps to record audio signals without setting static time for recording. The logic behind DLAR is described in Algorithm 1. As input to the algorithm, we provide a microphone stream that is in raw audio format, the number of features to analyze from the stream, the threshold for comparison of data at time-stamps, and maximum silence time until recording is stopped. As an output, we get audio data that is created from the stream. Going deep into Algorithm 1, while recording audio (line 6), for every iteration of the while loop that is being executed every 0.02 seconds, we calculate the spectrum average [45] of a small chunk of audio data from the microphone stream (lines 7-8) and compare the difference of the current spectrum average with the spectrum average of the first chunk (line 13). In case when the calculated difference is less than the given threshold (line 14), we consider that silence occurred and the silence counter is increased (line 15), otherwise the counter is set to 0 (line 21). Whenever the silence will reach the maximum silence time (line 16), the recording will stop, and audio data from the stream will be created (lines 17-18). In this work, we set the silence time to be 1.5 seconds so that DLAR listens to the user's speech request until 1.5-second silence occurs.

Algorithm 1 Dynamic Length Audio Recording**Input:** *MicStream*, *NofFeats*, *Thresh*, *MaxSilence***Output:** *AudioData*

```

1: Recording  $\leftarrow$  True
2: FirstLog  $\leftarrow$  True
3: FirstSA  $\leftarrow$  Null
4: SilenceCounter  $\leftarrow$  0
5: AudioData  $\leftarrow$  Null
6: while Recording do
7:   S  $\leftarrow$  GetSpectrum(MicStream, NofFeats)
8:   SA  $\leftarrow$  Average(S)
9:   if FirstLog then
10:    FirstSA  $\leftarrow$  SA
11:    FirstLog  $\leftarrow$  False
12:   else
13:    Diff  $\leftarrow$  GetDiff(FirstSA, SA)
14:    if Diff < Thresh then
15:      SilenceCounter  $\leftarrow$  SilenceCounter + 1
16:      if SilenceCounter = MaxSilence then
17:        AudioData  $\leftarrow$  Stop(MicStream)
18:        Recording  $\leftarrow$  False
19:      end if
20:    else
21:      SilenceCounter  $\leftarrow$  0
22:    end if
23:  end if
24: end while
25: return AudioData

```

Comparing the proposed dynamic approach DLAR with the static one, the recording time using DLAR is 192,632.74 seconds for possible 45,244 requests, whereas, with the static approach we got 344,306.84 seconds. In the case of DLAR, the wasting time for all requests is 1.5 seconds, whereas the static approach showed on average 4.85 seconds. Evaluating overall time efficiency, DLAR is 44.05 % more efficient than the static approach, which considerably speeds up inference and response time of the system.

Once the speech signal of the request is converted to audio data it is passed to an automatic speech recognizer model for further processing.

C. SPEECH-TO-TEXT

Speech-To-Text is an Automatic Speech Recognition neural network that takes audio data and extracts the spoken text from speech. It is based on the wav2vec2.0 network proposed in [31]. Speech-To-Text plays a vital role in the system since it creates a transcript text of an audio signal, and the performance of the model directly affects the next stages of inference in NSSE.

Regarding architecture, the wav2vec2.0 framework accepts a raw waveform of the speech signal and produces representations to be processed by Connectionist Temporal Classification [32] to create a transcript of the signal. The

model “encodes speech audio via a multi-layer convolutional neural network and then masks spans of the resulting latent speech representations similar to masked language modeling” [31], to be contextualized later using Transformers, where self-attention mechanism finds relationships in the sequence of latent representations in an end-to-end manner.

In the case of Aircraft Maintenance Metaverse, it is essential to note that there exists a professional vocabulary that contains aircraft maintenance-specific words and terms, therefore, existing models of wav2vec2.0 trained on general datasets might not work properly, even though they are in English (see Evaluation section). However, we assumed that it is more effective to fine-tune a pre-trained model, rather than training wav2vec2.0 from scratch, since ASR tasks require a vast amount of data. Therefore, in order to create Speech-To-Text in NSSE, we fine-tuned wav2vec2.0 which is pre-trained on general datasets such as Libri Speech [46], overcoming the issues of enormous datasets collection. We described training details and data in the Evaluation section.

Next, a created transcript of speech request in the form of text is passed to the Text-To-Programs for corresponding processing.

D. TEXT-TO-PROGRAMS

The Text-To-Programs component of NSSE is a deep learning sequence-to-sequence model that converts a text of a spoken command into a series of programs. In the system, a Program is a notation for a certain piece of code, a function that has its own parameters to be passed. Therefore, the main intuition behind Text-To-Programs is a translation of a text of request to a sequence of machine functions with parameters to be executed.

Fig. 4 describes the idea in detail. The system has knowledge of General Vocabulary, which is the words from possible users’ requests, and Domain-Specific Language that represents machine known words such as existing functions and possible parameters to be used for programs construction. Therefore, request text is converted into a Request Vector with the help of General Vocabulary that matches words from the text to the ones from a training dataset. Next, Text-To-Programs converts Request Vector to Programs Vector, which has references to the components of Domain-Specific Language to be used to generate programs. Thus, an example request text “Show AMM manual of item 8” is converted into programs “FindObject(Request)” and “ShowManual(AMM, Prev)”.

The architecture of Text-To-Programs is based on Transformers, which have an encoder-decoder type structure and amazingly suits translation tasks. Fig. 5 illustrates the architecture of Text-To-Programs models. Having word Embedding layers with dimension 256 both for Request Text input and Programs input, both embedding vectors combined with each word’s positional information in the form of positional encoding before being fed to encoders and decoders. In this work, the architecture consists of stacked 3 identical encoders, which map sequences into representations of the

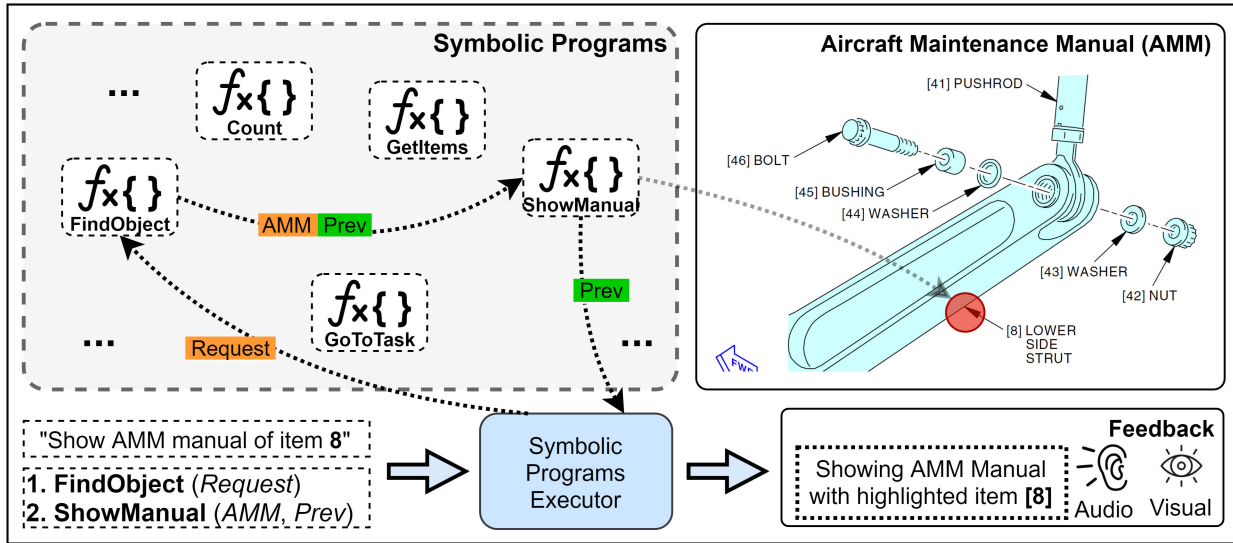


FIGURE 6. Symbolic programs executor working principle.

learned information for the given sequence, and 3 respective decoders, which generate text sequences, worked best with 8 multi-headed attention layers. Overall, the Request vocabulary size is 89, whereas, Programs vocabulary dimension is 49. Transformer’s output passed from Dropout 0.3 and Fully connected layer with no activation to get output probabilities.

Comparing our approach with [9] in terms of architecture, we applied Transformers in contrast to LSTM, which has certain issues. First, LSTMs are inefficient in speed, because to generate embeddings for a particular item in a sequence, the representations of every single word before have to be calculated, therefore, the computation process cannot be parallelized for running on GPUs. On contrary, the Transformer model can be trained and executed across multiple GPUs using parallelism pipeline. Second, LSTM lacks contextualization, due to comprehension of the meaning of a token according to the tokens that come before it but not the ones that come after. However, in Transformers every single token in a sequence is merged with every other token in that sequence at the same time, making context to be solid.

Finally, generated programs are passed the last part of NSSE processing for execution.

E. SYMBOLIC PROGRAMS EXECUTOR

Symbolic Programs Executor is a component in NSSE responsible for the execution of generated programs from the Text-To-Programs network and providing visual and audio feedback to users.

Algorithm 2 describes the process of Symbolic Programs Execution. As an input, the algorithm takes programs that have to be run. Each program consists of a function and its corresponding parameters. Entering the iteration (line 2), for each program in given programs, the function, and parameters are being extracted (lines 3-4). Next, variable `Prev`, which describes the result of the previous iteration, is appended to the parameters. Once functions and parameters are ready,

Algorithm 2 Symbolic Programs Executor

Input: *GivenPrograms*

Output: *Result*

- 1: $Prev \leftarrow Null$
- 2: **for** *Program*: *GivenPrograms* **do**
- 3: $Function \leftarrow Program.Function$
- 4: $Parameters \leftarrow Program.Parameters + Prev$
- 5: $Prev \leftarrow Execute(Function, Parameters)$
- 6: **end for**
- 7: $Result \leftarrow Prev$
- 8: **return** *Result*

Execute function invokes the respective function and passes extracted parameters (line 5). Each function has a return value, thus, in each iteration, variable `Prev` is updated (line 5). The above procedure is applied for all programs, where the last value of `Prev` describes the overall result from the execution (line 7). Return types of functions are different and created according to the needs.

Considering the example “Show me AMM manual of item 8” in Fig. 6, the corresponding programs are “FindObject(Request)” and “ShowManual(AMM, Prev)”. In this case, we have 2 programs that need to be executed sequentially by Symbolic Programs Executor. In the system, there exists Symbolic Programs space, which consists of machine code, functions that represent symbols, and Symbolic Program Executor matches generated programs to instances in machine space to invoke execution. In Fig 6, first, the FindObject function is called, it takes the Request parameter, which represents the transcript of the command (the result from the Speech-To-Text network). FindObject is a function that finds a number from the given text and returns it, thus, after execution of this program, the `Prev` variable is equal to 8, since 8 is a number mentioned in the example. Second, with the return value of FindObject, the ShowManual function with

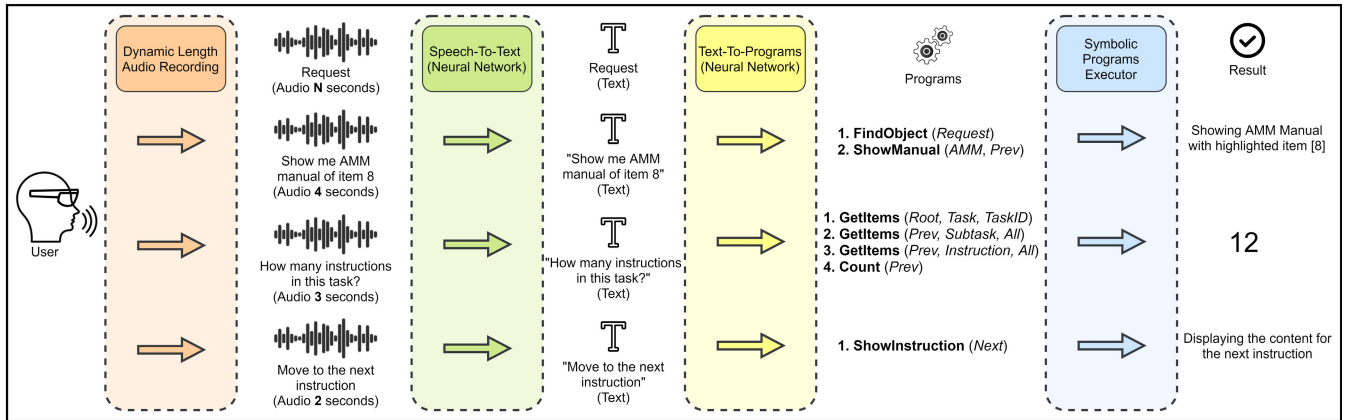


FIGURE 7. Sample user requests: from request to result.

parameters AMM and Prev is called. ShowManual is a function that displays a particular type of manual and highlights the number in it: in this case, the type of manual is AMM and the number to be highlighted is Prev, which holds the value 8 now (see Fig. 6). All functions in the domain-specific language have their own duty, some of them return processed computational operations, some perform validation and etc.

Once processing of all programs is finished, Symbolic Programs Executor processes feedback to users to enhance the user experience by notifying users about the ongoing procedure of displaying the manual both visually (in the form of text and icons) and vocally by playing i.e. “Showing AMM Manual with highlighted item 8”.

Combining components of NSSE all together, to better demonstrate inference, Fig. 7 with sample requests is provided. As can be seen, speech requests having various duration are handled by the Dynamic Length Audio Recording algorithm and transcribed by the Speech-To-Text model. Next, Text-To-Programs creates programs having various complexity and size. Taking the second example in the figure “How many instructions in this task?”, we may see that Text-To-Programs generated 4 interconnected programs, where the GetItems function gets information from the JSON knowledge file, extracting task with certain identification, next, from the previous result, NSSE extracts all subtasks since a task contains subtasks, and later, finds all instructions from subtasks because a subtask consists of instructions. Once all items are ready, which is the node with all instructions, the mathematical program Count will count items in the previous computed operation, giving the proper answer to the request of exact number of instructions.

The work of NSSE is based on Neuro-Symbolic AI, which combines the advantages of neural processing and symbolic reasoning to handle various contextual speech requests. Thus, the next section will analyze the importance of Neuro-Symbolic reasoning.

F. NEURO-SYMBOLIC REASONING OF NSSE

To handle speech requests from users and reply according to a specific context, it is effective to build a sys-

tem based on Neuro-Symbolic reasoning. When the neural components of NSSE perform complex pattern recognition in speech, the symbolic part manages the context and knowledge to give proper replies and validate users’ requests.

In Fig. 8, context management in NSSE is described. First, all legacy manuals such as AMM, IPC are structured in JSON format creating accessible and cross-referenced knowledge. This Aircraft Maintenance knowledge encapsulates all components, building relations, and dependencies. In the example Fig 8, various Task nodes have multiple Subtasks from AMM, and at the same time, Subtasks nodes with Instructions refer to aircraft-specific part numbers, such as items 51, 8, 42, and etc. all from manual, having their own 3D Models with Simulation procedures described in AMM. Regarding the certain state of operation, the required information is taken from JSON in addition to references to 3D virtual assets and Aircraft Maintenance knowledge into the Active State. This includes various environmental variables and links, such as current task, subtask and instruction information, available annotations in the AMM manual, 3D assets used in the current scene, simulations, etc. All these create the Context that has to be followed and reckoned with when NSSE handles speech commands.

Taking an example in Fig 8, Text-To-Programs neural network in NSSE, for requests “Show me AMM manual of item 8” and “Show me AMM manual of item 9” produces identical generated programs, however, Symbolic Programs Executor validates request according to the Context, available 3D assets, and overall knowledge, to give the final answer. Considering AMM items in Fig. 8 as a current context, the request of item 8 is valid, however, item 9 is not present in the AMM annotations, therefore, feedback to a user is corresponding.

In the neural part of semantics, when Text-To-Programs converts request text to machine-understandable programs, contextual information is not considered. Text-To-Programs tells to Symbolic Programs Executor what steps it needs to perform to get the result, however, symbolic reasoning happens while programs execution, which includes the

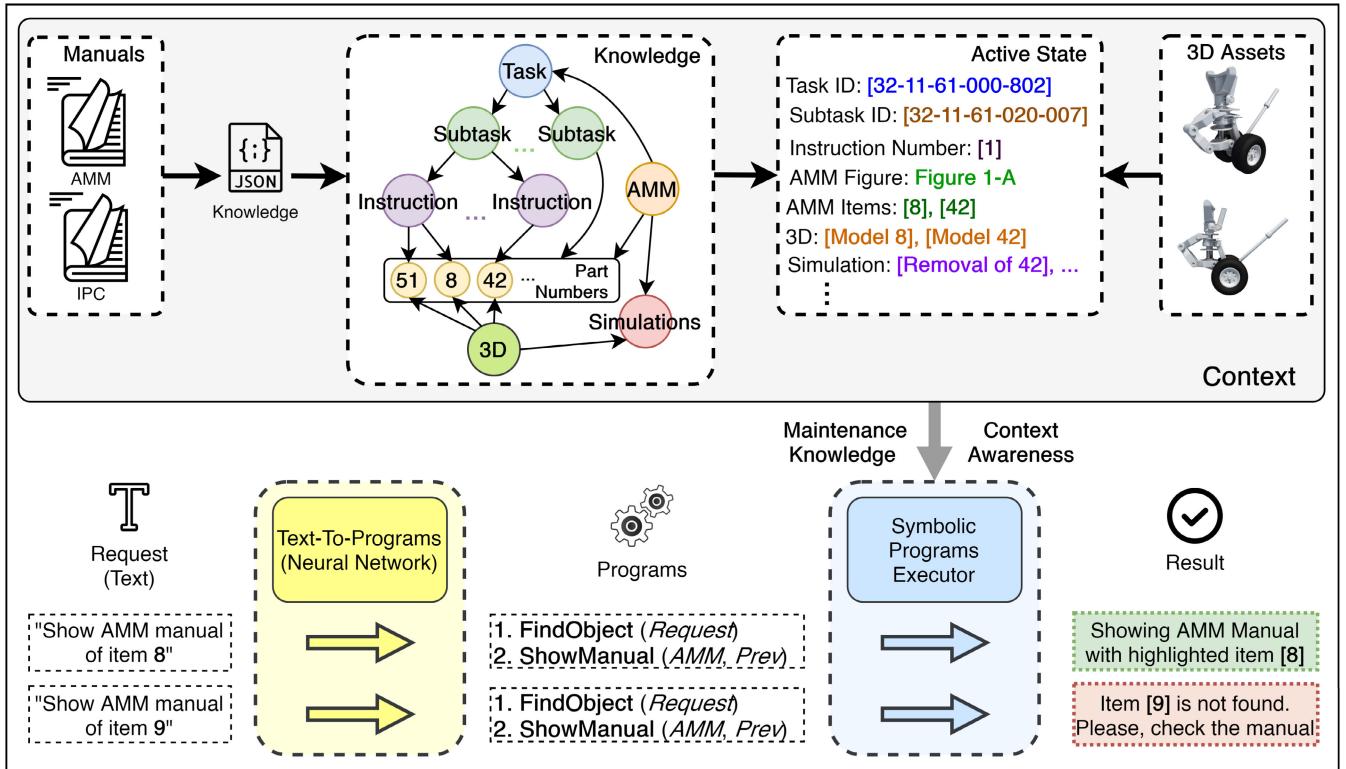


FIGURE 8. Context management in NSSE.

context-based validation procedure. Therefore, it is essential to have both neural and symbolic parts work together.

The next section describes the architecture of NSSE and the way components of the system communicate.

G. SYSTEM ARCHITECTURE

Neuro-Symbolic Speech Executor is a system that comprises four major components that are located across two devices, which follow Client-Server architecture. Fig. 9 describes the system architecture of NSSE, and as shown in the figure, there are a client machine, which is the user’s smart glasses, and a server, which is a deep learning machine that handles all processing. The client-side of the system is running on smart glasses HoloLens 2 and is responsible for creating speech requests and processing generated programs. Whereas, the server-side works with neural networks to convert audio data to text using Speech-To-Text, and text to programs with the help of Text-To-Programs. Two machines communicate through the internet, exchanging data: the client sends audio data and the server forwards transcript with generated programs back to the client. The following steps details procedure of inference:

1. Dynamic Length Audio Recording that is located on the client-side creates an audio request using the smart glasses microphone
2. Audio data is being sent over the web to the server machine
3. Received audio data is processed by the Speech-To-Text network to extract the transcript of the request

4. Text-To-Speech network converts transcript into a set of programs
5. Request text and generated programs is sent back to the client
6. Symbolic Programs Executor processes programs
7. Generated result with audio and visual feedback is demonstrated to the user.

This architecture guarantees that a client device - smart glasses is not overloaded with computational processing, since there are 2 neural networks in the system, without considering other 3D assets. Therefore, a powerful machine with GPU is installed to provide fast and efficient execution to process speech requests from devices. In addition, when the speech processing module is separated from devices, it can be easily maintained and updated, and provide services for applications built on different platforms such as smartphones, PC, tablets, etc.

Examples of tasks that are used in [9]–[11], considers the work of Neuro-Symbolic AI on research problems such as identifying what kind of a shape, color, relations a particular object has in the image, whereas in this work we demonstrated that the concept of Neuro-Symbolic AI can be applied in the industry to solve the real-world problem - handle complex semantic structure speech requests referencing contextual knowledge and environment. Comparing approaches, in [9] visual understanding and question answering is proposed, however, in our case, we process an audio signal to understand what the question is, and based on the given question functional programs are generated to be executed

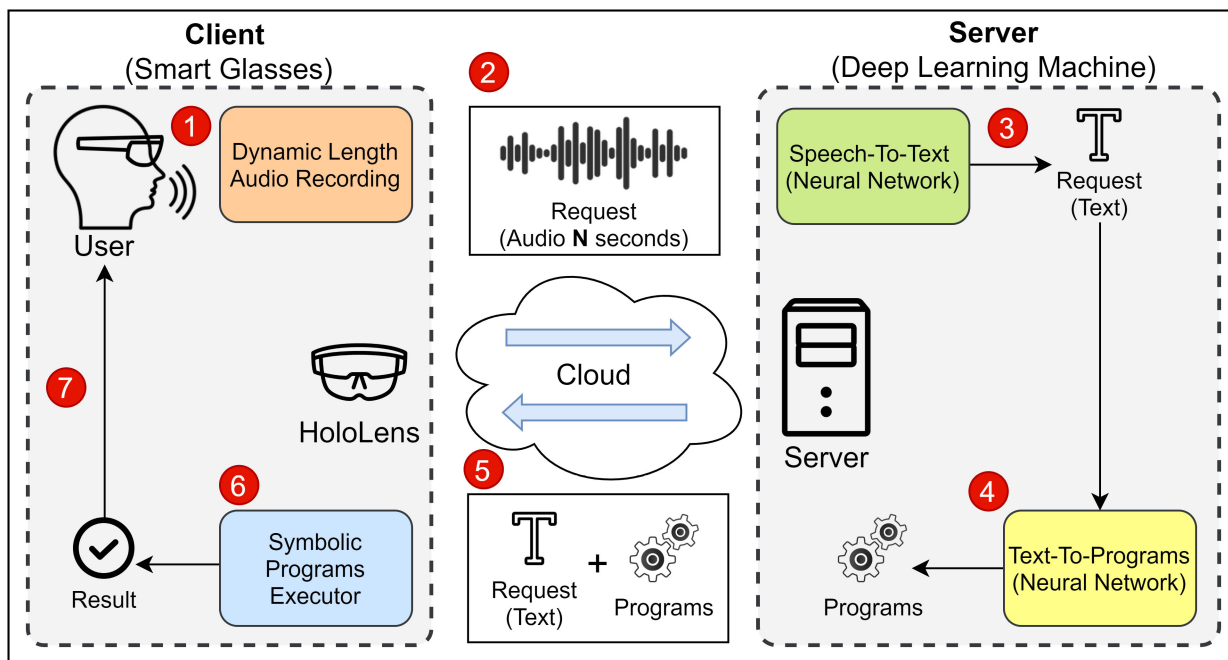


FIGURE 9. System architecture of neuro-symbolic speech executor.

on aircraft-related knowledge. What is more, in [9], an image is parsed to extract structural scene representation of visual data to perform execution of functional programs, however, we created such representation in the form of the JSON file, which summarizes the knowledge from maintenance manuals. Nevertheless, both techniques guarantee transparency of the reasoning process, which gives an opportunity to trace various problems and find explainable reasons for them, which is crucial for systems used in industry.

H. CONTRIBUTIONS

The first contribution of this work includes building a next-generation collaborative virtual space called aircraft maintenance metaverse that potentially can revolutionize the training and education process in aviation colleges. The proposed metaverse includes all the required resources for the MRO of an aircraft such as legacy manuals, 3D models and simulations, aircraft knowledge, and established maintenance flow, it replaces physical aircraft for training with virtual ones, saving a huge amount of resources. What is more, due to lack of resources, colleges use outdated models of aircraft for education, however, with the metaverse, the newest up-to-date knowledge can be easily maintained.

Second, we proposed the enhancement for existing aircraft maintenance manuals by building a 3D simulator that enables new dimensions for legacy knowledge. 3D manuals replicate the 2D AMM manual, adding animations and the ability for controlled step-by-step execution. Usually, a figure presented in 2D manuals depicts information only in one perspective, they are static, the proposed 3D manual makes full-side observation and interaction possible.

Third, to navigate and control operational flow in the metaverse, interact with the 3D manual, speech communication named Neuro-Symbolic Speech Executor is proposed. The contribution of this work is pushing forward the concept of Neuro-Symbolic AI by building context-aware speech understanding that can reason based on aircraft maintenance knowledge.

V. EVALUATION

In this section, we described the evaluation of NSSE. The detailed description of training and testing data is provided. Based on the amount of data and provided hardware details, training and inference times are provided to show the speed of processing. Furthermore, evaluation metrics and assessment strategies are explained in addition to quantitative experiment results and their corresponding discussion.

A. DATA

In order to perform training and evaluation, datasets for train and test procedures should be made. Analyzing the needs of the system and the particular functionality NSSE should perform, the list of possible user commands and corresponding symbolic programs is created. There are about 1,400 requests to programs pairs, which include aircraft maintenance domain-specific vocabulary. Using this list we can create spoken speech requests and transcripts to train the Speech-To-Text model, and at the same time, having text and programs pair, train the Text-To-Programs.

One of the objectives of this work is to keep an industrially cheap and flexible approach to build NSSE. Therefore, instead of employing people to speak out sample commands to build training dataset, we used Speech synthesizing

services such as Google Cloud Text-To-Speech [47]. Following the list of possible commands, Google API generated audio files of spoken requests making a dataset to train the Speech-To-Text network. Overall, 45,244 audio files from 46 various artificial speakers from Western countries are created. With Google API, it is cheap to generate hours of audio, and what is more, whenever the functionality of NSSE grows, synthesizing data makes updating the system considerably flexible, since there is no need to rely on human force.

Out of 45,244 synthetic audio requests, we used 42,981 for training Speech-To-Text and the rest 2,263 applied for testing and denoted this set as Test API. Since we want to use the system in a real environment, it is required to assess it with human data as well. Therefore, for evaluation purposes only, not training, we collected a dataset of audio requests from real humans, containing 5,831 audio requests from 9 people and denoted this set as Test Real. It is important to note that these people are from Eastern countries, such as Korea and Uzbekistan, and have intermediate English skills and distinctive pronunciation in comparison to native speakers in training data. This makes the assessment of NSSE interesting and challenging since training and test data have different features.

Summarizing the section, we have the synthetic training dataset of 42,981 spoken requests, synthetic test data of 2,263 requests, called Test API, and real humans spoken audio data denoted as Test Real, which consists of 5,831 audio requests from 9 people.

B. HARDWARE AND SOFTWARE DETAILS

In order to create the Speech-To-Text network, we fine-tuned wav2vec2.0 pre-trained models on training data. It takes about 6 hours to complete the training process that includes about 20 epochs. Regarding the Text-To-Programs model, it is quick in training time with only 5 minutes required for 25 epochs. The following technical properties are used for PC for training, testing, and the deep learning server described in section 4.

- Operating System: Ubuntu 18.04.4 LTS 64-bit
- CPU: Intel® Core i3-8100 CPU @ 3.60GHz × 4
- RAM: 8 GB
- GPU: GeForce RTX 2080 Ti/PCIe/SSE2 11 GB
- Framework: PyTorch 1.5.0 [48]

In addition, the proposed application of aircraft maintenance metaverse is created using the cross-platform game engine Unity [49] and deployed to smart glasses Microsoft HoloLens 2.

Talking about the inference time of NSSE, from the point when a user makes a request that is processed by DLAR, to send audio data and receive a reply back (steps 2 to 7 in the System Architecture section) it takes on average 196.5 milliseconds, where on average 104.9 milliseconds allocated for Speech-To-Text execution, and 27.8 milliseconds for Text-To-Programs conversion, and the rest is taken by other operations and network transfer time. As can be seen, NSSE works

in real-time creating a seamless experience. Next, evaluation metrics and strategy are discussed in the next section.

C. EVALUATION METRICS

To evaluate the system, we applied metrics that help to assess the performance of Speech-To-Text and Text-To-Programs. Since Symbolic Program Executor infers based on contextual information, we assumed that all given speech requests in test sets are valid in context, therefore, generated programs' correctness of Text-To-Programs defines the overall performance of NSSE. It is crucial to note that the correctness of request handling depends on each component of the system, however, a processing stage that comes before has a significant impact on the next one, in the example, Speech-To-Text performance plays a considerable role in Text-To-Speech results.

In order to assess the Speech-To-Text model, which is an Automatic Speech Recognition network, the following metrics are used to validate the correctness of the signal to text conversion.

- Word Error Rate (WER) - indicates the percentage of words that are incorrectly predicted. Working at the word level, it is derived from the Levenshtein distance of the number of minimum operations needed to obtain the ground truth from prediction sequences. The lower the value, the better the performance, with WER 0 being a perfect score [50].
- Word Recognition Rate (WRR) - indicates the percentage of words that are correctly predicted. Similar to WER, but has the opposite meaning.
- Character Error Rate (CER) - similar to WER, however, operates on character level instead of a word, therefore, it is a percentage of characters that are incorrectly predicted [51].
- Match Error Rate (MER) - is the proportion of word matches that are errors or the probability of a given match being incorrect [52].
- Word Information Lost (WIL) - is a probabilistic approximation to the proportion of word information lost. WIL "measures the proportion of mapping sensitive word information communicated" [52], giving a score between 0 and 1.
- Word Information Preserved (WIP) - is the opposite of WIL measure, which gives intuition about the preserved level of word information [52].

Once requests transcripts are generated by the Speech-To-Text model, the Text-To-Programs will process them to create programs. In order to evaluate the Text-To-Programs network, we used the Accuracy metric (denoted as ACC column in experimental results), which computes a number of correct programs over all programs. In this case, we consider 100% match as a correct prediction, which means that generated programs components should be fully equal to the ground truth programs to be considered as true. Thus, the Accuracy metric for the Text-To-Programs assesses the overall

TABLE 1. Evaluation of pre-trained wav2vec2.0 models on test API data.

Pre-trained Model	WER	WRR	CER	MER	WIL	WIP	ACC
Base	1.000	0.000	1.000	1.000	1.000	0.000	0.000
Libri-100h	0.521	0.479	0.095	0.443	0.634	0.366	0.512
Libri-960h	0.454	0.546	0.068	0.388	0.560	0.440	0.597
Timit	0.600	0.401	0.159	0.523	0.738	0.262	0.377

TABLE 2. Evaluation of pre-trained wav2vec2.0 models on test real data.

Pre-trained Model	WER	WRR	CER	MER	WIL	WIP	ACC
Base	1.000	0.000	1.000	1.000	1.000	0.000	0.000
Libri-100h	0.800	0.200	0.249	0.669	0.867	0.133	0.283
Libri-960h	0.687	0.313	0.193	0.591	0.803	0.197	0.343
Timit	0.994	0.006	0.367	0.783	0.940	0.060	0.162

TABLE 3. Evaluation of fine-tuned wav2vec2.0 models on test API data.

Fine-tuned Model	WER	WRR	CER	MER	WIL	WIP	ACC
Base	0.000	1.000	0.000	0.000	0.000	1.000	1.000
Libri-100h	0.009	0.991	0.004	0.009	0.015	0.985	1.000
Libri-960h	0.009	0.991	0.004	0.009	0.015	0.985	1.000
Timit	0.018	0.982	0.005	0.018	0.032	0.968	0.987

TABLE 4. Evaluation of fine-tuned wav2vec2.0 models on test real data.

Fine-tuned Model	WER	WRR	CER	MER	WIL	WIP	ACC
Base	0.142	0.858	0.045	0.141	0.244	0.756	0.884
Libri-100h	0.247	0.753	0.093	0.239	0.384	0.617	0.823
Libri-960h	0.141	0.859	0.045	0.137	0.231	0.769	0.894
Timit	0.207	0.793	0.083	0.203	0.332	0.668	0.829

performance of consecutively executed models: Speech-To-Text and Text-To-Programs. Based on the presented metrics, we discussed experimental results in the next section.

D. RESULTS

In this section, the results of the experiments are presented. Using test data and evaluation metrics described in the previous sections, we assess various pre-trained and fine-tuned models of Speech-To-Text, and overall accuracy with Text-To-Programs.

There are various Speech-To-Text networks that are already trained using public datasets such as Libri Speech [46] and Timit [53]. These datasets consist of general language samples, however, in aircraft maintenance, there exists domain-specific language with various professional words. Therefore, networks pre-trained on general datasets do not perform well on domain-specific vocabulary, resulting in poor performance. To show that, we evaluated pre-trained networks on test sets, next, we finetuned them with training data that consists of domain-specific language examples and made evaluation again on the same test data to compare. Since we have two test sets (Test API and Test Real) and two types of models (pre-trained and fine-tuned), we demonstrated the statistics in 4 tables (Table 1-4), where the evaluation of pre-trained or fine-tuned models on test API or Real data is presented.

Out of experimented pre-trained models in Table 1 and Table 2, the wav2vec2.0 model trained on 960 hours of Libri Speech (Pre-trained Libri-960h) dataset shows the best performance, among others. The best pre-trained model shows

45.4% WER and 59.7% Accuracy (ACC) on Test API data, and 68.7% WER and 34.3% Accuracy on Real Test data. After being finetuned with domain-specific training data, the Fine-Tuned Libri-960h model in Table 3 and Table 4 shows 0.9% WER and 100% Accuracy on Test API data and 14.1% WER and 89.4% Accuracy on Real Test data. The experiment illustrates that after finetuning with domain-specific samples the Fine-tuned pre-trained models (excluding Base models) show on average 51.26% and 49.68% better performance on WER and Accuracy respectively for Test API set, and 62.87% and 58.59% better performance on WER and Accuracy for Real Test set. This summarizes that pre-trained models on general datasets do not perform well on samples with professional words, thus, need to be fine-tuned to achieve acceptable results.

The system we developed creates global value and brings people around the world together, thus, it should not be used only by native English speakers, but also users having intermediate English skills, therefore, we should expect that various English pronunciations are different from what we trained (British, American, Australian, etc.). To illustrate that our system is capable of generalizing to other accents, we created the Test Real set of audio samples recorded by non-native speakers from eastern countries, whose pronunciations have distinguishable features from native ones, and the level of English is intermediate. Along with Test API data, which can be considered as test data of western native speakers, we evaluated performances on Test Real and compare the results in Table 2 and Table 4. The best models on Test Real data show 14.1% of WER and 89.4% of overall Accuracy, which is about 10% lower in comparison with Test API. NSSE shows close to perfect performance on Test API data, which is from the same distribution as training data for finetuning and represents native speakers. At the same time, it can be seen that the system can handle requests from users having different pronunciations and lower English levels.

Summarizing evaluation, to create a speech communication system for the aircraft domain, speech recognition models should be trained with domain-specific vocabulary because existing models pre-trained on general datasets are not capable of handling professional words and commands with certain structures. Instead, it is efficient to apply transfer learning for pre-trained models by fine-tuning them with domain-specific data. After being assessed with test data and various metrics, experimental results demonstrated that on average NSSE achieved an accuracy of 94.7% and Word Error Rate of 7.5%. According to [12], [54], a system having WER 5-10 % is considered to be sustainable for automatic speech recognition tasks to be used in the industry. Overall, we can consider NSSE as a bot that acts as the aircraft domain expert who understands spoken language, answers questions, provides requested resources, and guides trainees toward task completion, and makes sure that the maintenance process is strictly followed by manuals since it is vital for safety.

VI. CONCLUSION

To conclude, in contemporary times, industries move towards online alternatives - metaverses to socialize people in virtual spaces, and having metaverse for aircraft maintenance training can provide an affordable solution for aviation colleges to have education on virtual aircraft instead of costly physical machines. Therefore, in this work, we proposed the Aircraft Maintenance Metaverse of Boeing-737 - the mixed reality environment of maintenance training and education that provides 3D aircraft-specific digital assets, legacy manuals, aircraft maintenance knowledge, and shared collaboration space that accessed with the help of smart glasses HoloLens 2. Complementing and enhancing currently used paper-based legacy manuals, we created 3D manuals that simulate operations on virtual models of aircraft just like on a physical one, allowing full-side observation, interactions, and step-by-step execution, opening new dimensions for legacy knowledge.

Moreover, in order to control the operational workflow and interact with digital assets within the metaverse, context-aware speech understanding module Neuro-Symbolic Speech Executor is proposed. The field of aircraft MRO is strictly followed by the manuals, however, conventional deep learning speech understanding approaches are not context-oriented. Therefore, the Neuro-Symbolic AI is applied to handle speech requests according to contextual information. NSSE combines speech recognition and language modeling neural networks with traditional symbolic AI to reason on aircraft maintenance knowledge. NSSE is trained using synthesized data, but it is still generalizable on real human speech requests of various native and non-native pronunciations. Experimental results demonstrated that NSSE is robust for industrial use, showing an average Word Error Rate of 7.5% and overall accuracy of 94.7%.

For future works, we plan to upgrade NSSE to differentiate millions of parts in aircraft maintenance knowledge. Currently, in order to refer to a certain item, its specific identification number is used. However, in the future, we want NSSE to refer and specify aircraft parts by multiple features such as item name, ID, description and etc. Since there is a huge amount of components and they differ from manual to manual, a dynamic and scalable approach needs to be developed to cope with this task. In addition, we plan to build multilingual support for NSSE and explore how context-aware reasoning mechanisms perform when having several languages.

ACKNOWLEDGMENT

The authors thank Augmented Knowledge Corporation (www.augmentedk.com) for providing resources for development and experimentation.

REFERENCES

- [1] *The Metaverse: A Brave, New (Virtual) World* | by Micaela Mantegna | Berkman Klein Center Collection | Jun, 2021 | Medium. Accessed: Mar. 8, 2021. [Online]. Available: <https://medium.com/berkman-klein-center/the-metaverse-a-brave-new-virtual-world-2f040cbae7d4>
- [2] *More Than a Trend: Entering the Metaverse Will Become a Necessity for Brands*. Accessed: Jun. 8, 2021. [Online]. Available: <https://www.forbes.com/sites/cathyhackl/2021/06/24/more-than-a-trend-e-entering-the-metaverse-will-become-a-necessity-for-brands>
- [3] *The Metaverse is Coming and it's a Very Big Deal*. Accessed: May 8, 2021. [Online]. Available: <https://www.forbes.com/sites/cathyhackl/2020/07/05/the-metaverse-is-coming-its-a-very-big-deal>
- [4] *Boeing: Next-Generation 737*. Accessed: Mar. 8, 2021. [Online]. Available: <https://www.boeing.com/commercial/737ng/>
- [5] *How Much do Boeing Aircraft Cost—Simple Flying*. Accessed: May 8, 2021. [Online]. Available: <https://simpleflying.com/how-much-do-boeing-aircraft-cost/>
- [6] *HoloLens 2—Overview, Features, and Specs* | Microsoft HoloLens. Accessed: Mar. 8, 2021. [Online]. Available: <https://www.microsoft.com/en-us/hololens/hardware>
- [7] *How to Use Aircraft Maintenance Manual* | Aviationhunt. Accessed: Mar. 8, 2021. [Online]. Available: <https://www.aviationhunt.com/aircraft-maintenance-manual/>
- [8] *Illustrated Parts Catalogue Skybrary Aviation Safety*. Accessed: Mar. 8, 2021. [Online]. Available: https://www.skybrary.aero/index.php/illustrated_parts_catalogue
- [9] K. Yi, J. Wu, C. Gan, A. Torralba, P. Kohli, and J. B. Tenenbaum, "Neural-symbolic VQA: Disentangling reasoning from vision and language understanding," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 1039–1050.
- [10] C. Han, J. Mao, C. Gan, J. Tenenbaum, and J. Wu, "Visual concept-metacognition learning," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, H. Wallach, H. Larochelle, A. Beygelzimer, F. d Alché-Buc, E. Fox, and R. Garnett, Eds. Red Hook, NY, USA: Curran Associates, Inc., 2019, pp. 5001–5012. [Online]. Available: <https://proceedings.neurips.cc/paper/2019/file/98d8a23fd60826a2a474c5b4%2f5811707-Paper.pdf>
- [11] J. Mao, C. Gan, P. Kohli, J. B. Tenenbaum, and J. Wu, "The neuro-symbolic concept learner: Interpreting scenes, words, and sentences from natural supervision," in *Proc. 7th Int. Conf. Learn. Represent., (ICLR)*, New Orleans, LA, USA, May 2019, pp. 1–28. [Online]. Available: <https://openreview.net/forum?id=rJgMlhRctm>
- [12] *Evaluate and Improve Custom Speech Accuracy—Speech Service—Azure Cognitive Services* | Microsoft Docs. Accessed: Mar. 8, 2021. [Online]. Available: <https://docs.microsoft.com/en-us/azure/cognitive-services/speech-service/how-to-custom-speech-evaluate-data>
- [13] M. R. Mirzaei, S. Ghorshi, and M. Mortazavi, "Combining augmented reality and speech technologies to help deaf and hard of hearing people," in *Proc. 14th Symp. Virtual Augmented Reality*, May 2012, pp. 174–181, doi: 10.1109/SVR.2012.10.
- [14] I. Dabran, T. Avny, E. Singher, and H. Ben Danan, "Augmented reality speech recognition for the hearing impaired," in *Proc. IEEE Int. Conf. Microw., Antennas, Commun. Electron. Syst. (COMCAS)*, Nov. 2017, pp. 1–4, doi: 10.1109/COMCAS.2017.8244731.
- [15] A. Virkkunen, "Automatic speech recognition for the hearing impaired in an augmented reality application," M.S. thesis, School Sci., Aalto Univ., Espoo, Finland, 2018. [Online]. Available: <http://urn.fi/URN:NBN:fi:aalto-201812146531>
- [16] M. R. Mirzaei, S. Ghorshi, and M. Mortazavi, "Audio-visual speech recognition techniques in augmented reality environments," *Vis. Comput.*, vol. 30, no. 3, pp. 245–257, Mar. 2014.
- [17] C. S. Che Dalim, M. S. Sunar, A. Dey, and M. Billinghurst, "Using augmented reality with speech input for non-native children's language learning," *Int. J. Hum.-Comput. Stud.*, vol. 134, pp. 44–64, Feb. 2020. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1071581918303161>
- [18] M. Majewski and W. Kacalak, "Conceptual design of innovative speech interfaces with augmented reality and interactive systems for controlling loader cranes," *Adv. Intell. Syst. Comput.*, vol. 464, pp. 237–247, Apr. 2016.
- [19] A. Siyaev and G.-S. Jo, "Towards aircraft maintenance metaverse using speech interactions with virtual objects in mixed reality," *Sensors*, vol. 21, no. 6, p. 2066, Mar. 2021.
- [20] *Voice Input Mixed Reality* | Microsoft Docs. Accessed: Oct. 18, 2021. [Online]. Available: <https://docs.microsoft.com/en-us/windows/mixed-reality/design/voice-input>
- [21] Q. H. Nguyen and T.-D. Cao, "A novel method for recognizing Vietnamese voice commands on smartphones with support vector machine and convolutional neural networks," *Wireless Commun. Mobile Comput.*, vol. 2020, pp. 1–9, Mar. 2020.

- [22] P. Jansson, "Single-word speech recognition with convolutional neural networks on raw waveforms," M.S. thesis, Inf. Technol., Arcada Univ., Espoo, Finland, 2018. [Online]. Available: https://www.theseus.fi/bitstream/handle/10024/144982/Jansson_Patrick.p%df
- [23] R. A. Solovyev, M. Vakhrushev, A. Radionov, I. I. Romanova, A. A. Amerikanov, V. Aliev, and A. A. Shvets, "Deep learning approaches for understanding simple speech commands," in *Proc. IEEE 40th Int. Conf. Electron. Nanotechnol. (ELNANO)*, Apr. 2020, pp. 688–693.
- [24] R. Sharmin, S. K. Rahut, and M. R. Huq, "Bengali spoken digit classification: A deep learning approach using convolutional neural network," *Proc. Comput. Sci.*, vol. 171, pp. 1381–1388, Jan. 2020. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1877050920311273>
- [25] L. Nanni, G. Maguolo, S. Brahmam, and M. Paci, "An ensemble of convolutional neural networks for audio classification," *Appl. Sci.*, vol. 11, no. 13, p. 5796, Jun. 2021. [Online]. Available: <https://www.mdpi.com/2076-3417/11/13/5796>
- [26] D. Amodei *et al.*, "Deep speech 2: End-to-end speech recognition in english and Mandarin," 2016, *arXiv:1512.02595*.
- [27] S. Schneider, A. Baevski, R. Collobert, and M. Auli, "wav2vec: Unsupervised pre-training for speech recognition," in *Proc. Interspeech*, Sep. 2019, pp. 1–9.
- [28] G. Synnaeve, Q. Xu, J. Kahn, T. Likhomanenko, E. Grave, V. Pratap, A. Sriram, V. Liptchinsky, and R. Collobert, "End-to-end ASR: From supervised to semi-supervised learning with modern architectures," 2019, *arXiv:1911.08460*.
- [29] T. Hori, S. Watanabe, Y. Zhang, and W. Chan, "Advances in joint CTC-attention based end-to-end speech recognition with a deep CNN encoder and RNN-LM," in *Proc. Interspeech*, Aug. 2017, pp. 4835–4839.
- [30] C.-C. Chiu, T. N. Sainath, Y. Wu, R. Prabhavalkar, P. Nguyen, Z. Chen, A. Kannan, R. J. Weiss, K. Rao, E. Gonina, N. Jaitly, B. Li, J. Chorowski, and M. Bacchiani, "State-of-the-art speech recognition with sequence-to-sequence models," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2018, pp. 4774–4778.
- [31] A. Baevski, H. Zhou, A. Mohamed, and M. Auli, "Wav2vec 2.0: A framework for self-supervised learning of speech representations," 2020, *arXiv:2006.11477*.
- [32] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks," in *Proc. 23rd Int. Conf. Mach. Learn. (ICML)*, 2006, pp. 369–376.
- [33] *Neuro-Symbolic AI IBM*. Accessed: Mar. 8, 2021. [Online]. Available: https://researcher.watson.ibm.com/researcher/view_group.php?id=10518
- [34] M. K. Sarker, L. Zhou, A. Eberhart, and P. Hitzler, "Neuro-symbolic artificial intelligence: Current trends," 2021, *arXiv:2105.05330*.
- [35] J. Johnson, B. Hariharan, L. van der Maaten, L. Fei-Fei, C. L. Zitnick, and R. Girshick, "CLEVR: A diagnostic dataset for compositional language and elementary visual reasoning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1988–1997.
- [36] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [37] L. D. Raedt, S. Dumančić, R. Manhaeve, and G. Marra, "From statistical relational to neuro-symbolic artificial intelligence," in *Proc. 29th Int. Joint Conf. Artif. Intell.*, Jul. 2020, pp. 4943–4950.
- [38] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. U. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Red Hook, NY, USA: Curran Associates, 2017, pp. 5998–6008. [Online]. Available: <https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fb0d53c1%c4a845aa-Paper.pdf>
- [39] *Google AI Blog: Transformer: A Novel Neural Network Architecture for Language Understanding*. Accessed: Mar. 8, 2021. [Online]. Available: <https://ai.googleblog.com/2017/08/transformer-novel-neural-network.htm%l>
- [40] S. Karita, N. Chen, T. Hayashi, T. Hori, H. Inaguma, Z. Jiang, M. Someki, N. E. Y. Soplin, R. Yamamoto, X. Wang, S. Watanabe, T. Yoshimura, and W. Zhang, "A comparative study on transformer vs RNN in speech applications," in *Proc. IEEE Autom. Speech Recognit. Understand. Workshop (ASRU)*, Dec. 2019, pp. 449–456.
- [41] A. Sherstinsky, "Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network," 2018, *arXiv:1808.03314*.
- [42] *Metaverse Wikipedia*. Accessed: Aug. 4, 2021. [Online]. Available: <https://en.wikipedia.org/wiki/Metaverse>
- [43] S.-N. Suzuki, H. Kanematsu, D. M. Barry, N. Ogawa, K. Yajima, K. T. Nakahira, T. Shirai, M. Kawaguchi, T. Kobayashi, and M. Yoshitake, "Virtual experiments in metaverse and their applications to collaborative projects: The framework and its significance," *Proc. Comput. Sci.*, vol. 176, pp. 2125–2132, Jan. 2020.
- [44] *The Metaverse: What it is, Where to Find it, Who Will Build it, and Fortnite Matthewball.Vc*. Accessed: Apr. 8, 2021. [Online]. Available: <https://www.matthewball.vc/all/themetaverse>
- [45] *Unity Scripting Api: AudioSource.GetSpectrumData*. Accessed: Mar. 8, 2021. [Online]. Available: <https://docs.unity3d.com/ScriptReference/AudioSource.GetSpectrumData.h%tml>
- [46] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An ASR corpus based on public domain audio books," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2015, pp. 5206–5210.
- [47] *Text-to-Speech: Lifelike Speech Synthesis | Google Cloud*. Accessed: Mar. 8, 2021. [Online]. Available: <https://cloud.google.com/text-to-speech>
- [48] *Pytorch*. Accessed: Mar. 8, 2021. [Online]. Available: <https://pytorch.org/>
- [49] *Unity Real-Time Development Platform | 3D, 2D VR & AR Engine*. Accessed: Aug. 13, 2021. [Online]. Available: <https://unity.com/>
- [50] A. Ali and S. Renals, "Word error rate estimation for speech recognition: E-WER," in *Proc. 56th Annu. Meeting Assoc. Comput. Linguistics (Short Papers)*, vol. 8, 2018, pp. 20–24.
- [51] T. Wolf, J. Chaumond, L. Debut, V. Sanh, C. Delangue, A. Moi, P. Cistac, M. Funtowicz, J. Davison, S. Shleifer, and R. Louf, "Transformers: State-of-the-art natural language processing," in *Proc. Conf. Empirical Methods Natural Lang. Process., Syst. Demonstrations*. Stroudsburg, PA, USA: Association for Computational Linguistics, Oct. 2020, pp. 38–45. [Online]. Available: <https://www.aclweb.org/anthology/2020.emnlp-demos.6>
- [52] A. C. Morris, V. Maier, and P. Green, "From WER and RIL to MER and WIL: Improved evaluation measures for connected speech recognition," in *Proc. Interspeech*, Oct. 2004, pp. 1–4.
- [53] J. Garofolo, L. Lamel, W. Fisher, J. Fiscus, D. Pallett, and N. Dahlgren, "Darpa limit acoustic-phonetic continuous speech corpus cd-rom TIMIT," Nat. Inst. Standards Technol., Gaithersburg, MD, USA, NIST Interagency/Internal Rep. 4930, Feb. 1993.
- [54] *What is Wer? What Does Word Error Rate Mean? Rev*. Accessed: Mar. 8, 2021. [Online]. Available: <https://www.rev.com/blog/resources/what-is-wer-what-does-word-error-ra%te-mean>



AZIZ SIYAEV received the B.S. degree in computer science and engineering from Inha University in Tashkent (IUT), Tashkent, Uzbekistan, in 2018, and the M.S. degree in electrical and computer engineering from Inha University, Incheon, South Korea, in 2020, where he is currently pursuing the Ph.D. degree in electrical and computer engineering.

Since 2018, he has been a Graduate Research Assistant with the Artificial Intelligence Laboratory, Inha University. His research interests include neuro-symbolic AI, mixed reality, speech recognition, and generative adversarial networks.



GEUN-SIK JO (Senior Member, IEEE) received the B.S. degree in computer science from Inha University, Incheon, South Korea, in 1982, the M.S. degree in computer science from the Queens College, City University of New York, New York City, NY, USA, in 1985, and the Ph.D. degree in computer science from the City University of New York, in 1991.

Since 1991, he has been a Professor with the Electrical and Computer Engineering Department, Inha University. Since 2016, he has been working as the Head of the AI Contents Creation Research Center, Inha University. Since 2017, he has been a CEO and a Founder of Augmented Knowledge Corporation. He is the author of multiple books, more than 300 articles, and 52 patents. His research interests include augmented reality, deep learning, ontologies, knowledge-based systems, and semantic web.

...