aerospace

MDPI

Article

In-Vehicle Speech Recognition for Voice-Driven UAV Control in a Collaborative Environment of MAV and UAV

Jeong-Sik Park [1,*] and Na Geng [2]

[1] Department of English Linguistics and Language Technology, Hankuk University of Foreign Studies, Seoul 02450, Republic of Korea
[2] Department of English Linguistics, Hankuk University of Foreign Studies, Seoul 02450, Republic of Korea; gengna0324@gmail.com
* Correspondence: parkjs@hufs.ac.kr; Tel.: +82-02-2173-8814

Abstract: Most conventional speech recognition systems have mainly concentrated on voice-driven control of personal user devices such as smartphones. Therefore, a speech recognition system used in a special environment needs to be developed in consideration of the environment. In this study, a speech recognition framework for voice-driven control of unmanned aerial vehicles (UAVs) is proposed in a collaborative environment between manned aerial vehicles (MAVs) and UAVs, where multiple MAVs and UAVs fly together, and pilots on board MAVs control multiple UAVs with their voices. Standard speech recognition systems consist of several modules, including front-end, recognition, and post-processing. Among them, this study focuses on recognition and post-processing modules in terms of in-vehicle speech recognition. In order to stably control UAVs via voice, it is necessary to handle the environmental conditions of the UAVs carefully. First, we define control commands that the MAV pilot delivers to UAVs and construct training data. Next, for the recognition module, we investigate an acoustic model suitable for the characteristics of the UAV control commands and the UAV system with hardware resource constraints. Finally, two approaches are proposed for post-processing: grammar network-based syntax analysis and transaction-based semantic analysis. For evaluation, we developed a speech recognition system in a collaborative simulation environment between a MAV and an UAV and successfully verified the validity of each module. As a result of recognition experiments of connected words consisting of two to five words, the recognition rates of hidden Markov model (HMM) and deep neural network (DNN)-based acoustic models were 98.2% and 98.4%, respectively. However, in terms of computational amount, the HMM model was about 100 times more efficient than DNN. In addition, the relative improvement in error rate with the proposed post-processing was about 65%.

Keywords: speech recognition; voice-driven control; acoustic model; grammar network; syntax analysis; semantic analysis; unmanned aerial vehicle (UAV); UAV control

check for updates

Citation: Park, J.-S.; Geng, N. In-Vehicle Speech Recognition for Voice-Driven UAV Control in a Collaborative Environment of MAV and UAV. *Aerospace* **2023**, *10*, 841. https://doi.org/10.3390/aerospace10100841

Academic Editor: Peng Wei

Received: 18 August 2023
Revised: 11 September 2023
Accepted: 26 September 2023
Published: 27 September 2023

Correction Statement: This article has been republished with a minor change. The change does not affect the scientific content of the article and further details are available within the backmatter of the website version of this article.

1. Introduction

Since speech recognition technology has been successfully used in personal assistant devices such as artificial intelligence (AI) speakers and smartphones, various speech recognition applications have been introduced. In particular, many attempts have been made to apply voice control to moving objects such as cars, and the speech recognition function has played a very important role in controlling flying objects such as unmanned aerial vehicles (UAVs). In order to control a moving object through speech recognition in such a special environment, research considering the specificity of the environment is necessary. This study proposes a speech recognition framework for voice-based UAV control in a collaborative environment of manned aerial vehicles (MAVs) and UAVs. In this environment, multiple MAVs and multiple UAVs fly together, and pilots on board the MAVs perform collaborative tasks with UAVs by controlling multiple UAVs with their voices.

Several previous studies introduced systems for controlling UAVs via voice [1–3]. Most conventional studies have focused on a typical speech recognition environment where speech recognition controls a single UAV. In a previous study, we presented an efficient speech recognition architecture and front end for controlling multiple UAVs with voice [4].

If the current speech recognition scheme (that is, server-centric scheme) that processes speech recognition in a remote server is applied to a collaborative environment of a MAV and an UAV, various problems may arise. First, multiple MAV pilots simultaneously submitting voice commands to a single server can place a heavy burden on the server, delaying the sending of commands. The server-centric scheme also manages indirect UAV control by performing three data transmission sequences: the MAV to the recognition server, the server to the MAV, and the MAV to the UAV. This indirect communication can incur communication costs, resulting in misrecognition or dropped commands due to packet loss while the MAV and the UAV are moving. For high-speed moving, special-purpose UAVs (e.g., military UAVs), the packet loss problem can be more serious.

In [4], we proposed an efficient recognition scheme to solve such disadvantages of conventional speech recognition schemes in the multi-UAV control via voice. The proposed scheme is summarized as distributed speech recognition in which the MAV and UAV share speech recognition processes. The MAV system processes the front-end module to extract acoustic features from the input speech uttered by the MAV pilot. When the acoustic features are sent to the UAVs, the UAV's system performs the recognition process that follows the front-end process.

This study introduces a speech recognition process not covered in the previous study. There are few studies considering an efficient speech recognition framework for the environment where a MAV and an UAV cooperate to perform military operations. This study proposes a speech recognition framework suitable for this environment. In particular, we concentrate on an acoustic model suitable for a distributed speech recognition system in which the MAV and UAV share speech recognition tasks and a post-processing method to minimize the risk caused by speech recognition errors in a collaborative environment of a MAV and an UAV.

In recent years, research on speech recognition and understanding in air traffic control (ATC) environments have been conducted through projects such as HAAWAII [5] and SESAR [6]. In a voice communication environment between air traffic controllers (ATCo) and the pilot, the ATCo receives the pilot's voice command and performs recognition. ATC systems with relatively high-performance hardware can handle complex models such as end-to-end ASR models [7–9]. However, in the collaborative environment between an UAV and a MAV targeted in this study, the speech recognition module is operated on the UAV with hardware resource constraints. Thus, the conventional research has somewhat different characteristics from the environment we are targeting in that the ATC system has relatively few hardware resource limitations and is processed at the ground control center.

The remainder of this paper is organized as follows. In Section 2, we propose an efficient speech recognition framework for voice-driven UAV control in a collaborative environment of MAVs and UAVs. In Section 3, several experiments conducted on speech data and their results are reported and discussed. Finally, Section 4 concludes the paper.

2. In-Vehicle Speech Recognition for Voice-Driven UAV Control in a Collaborative Environment of MAV and UAV

Although considerable research on speech recognition has been conducted in various fields, research on speech recognition for UAV control is relatively insufficient [10]. Several studies on multimodality using speech and visual data have been introduced [10,11], and most of the speech recognition studies have considered situations where speech recognition is performed at a ground control station [12–14]. However, most speech recognition processes for UAV control are performed in a similar way to general speech recognition systems.

A traditional speech recognition system consists of several modules, including front-end, recognition, and post-processing modules, as shown in Figure 1 [15,16]. The front-end module performs several processes, such as noise reduction, voice triggering, and acoustic feature extraction. Next, in the recognition module, speech recognition is performed using pre-trained acoustic models using common pattern recognition techniques such as deep neural networks (DNNs) or hidden Markov models (HMMs). Finally, the post-processing module performs syntax and semantic analyses to improve the recognition output's accuracy and clarity.

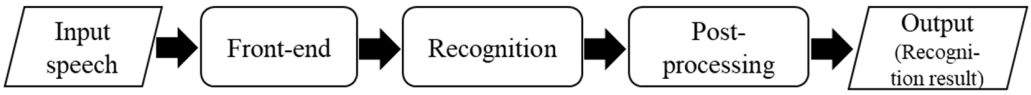

Figure 1. The general procedure of standard speech recognition systems.

As mentioned in Section 1, a speech recognition scheme suitable for the collaborative environment of MAVs and UAVs is a form of distributed speech recognition in which a MAV and an UAV share speech recognition processes. Figure 2 summarizes this scheme in which the MAV processes the front-end module and the UAV performs the recognition process.

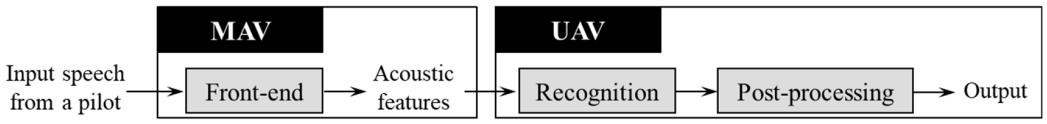

Figure 2. Distributed speech recognition for the collaborative environment of MAVs and UAVs.

In this section, we propose each module suitable for collaborative environments of MAVs and UAVs.

2.1. Front-End of Speech Recognition

The front end of speech recognition consists of four main processes: voice activity detection (VAD), feature extraction, noise reduction, and voice trigger, as shown in Figure 3 [4,17]. The first two processes are essential for speech recognition. VAD is the process of detecting target speech regions to perform speech recognition. Feature extraction is to extract features representing acoustic characteristics in the time or frequency domain from input speech data.

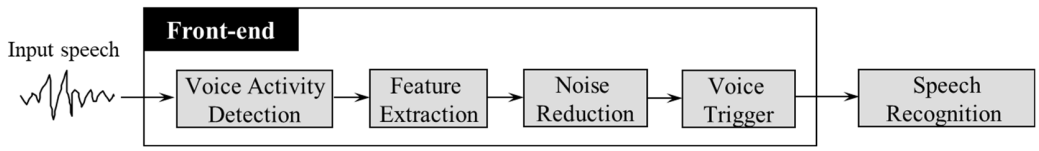

Figure 3. The general procedure of front-end speech recognition.

Noise reduction and voice triggering should be developed according to the system environment. In the previous study, noise reduction and voice-triggering approaches were proposed to handle multi-UAV environments [4]. In particular, to consider multi-UAV control, we proposed a multi-channel voice-trigger approach in which each UAV has a unique name used as a trigger word, and the MAV pilot establishes a connection between the MAV and the target UAV among multiple UAVs. Figure 4 represents the multi-channel voice trigger-based front end and speech recognition procedures for multi-UAV control. When MAV pilots have a conversation and a situation arises where they need to call an UAV among multiple UAVs, they call the name corresponding to the target UAV. Then, the voice-trigger module detects a specific UAV name according to the process shown in the upper block of the figure, and it attempts to connect with the target UAV. When connected to the target UAV, the pilot speaks a command, and the features extracted from the voice

are transmitted to the target UAV, and finally, speech recognition proceeds, as shown in the block below in Figure 4.

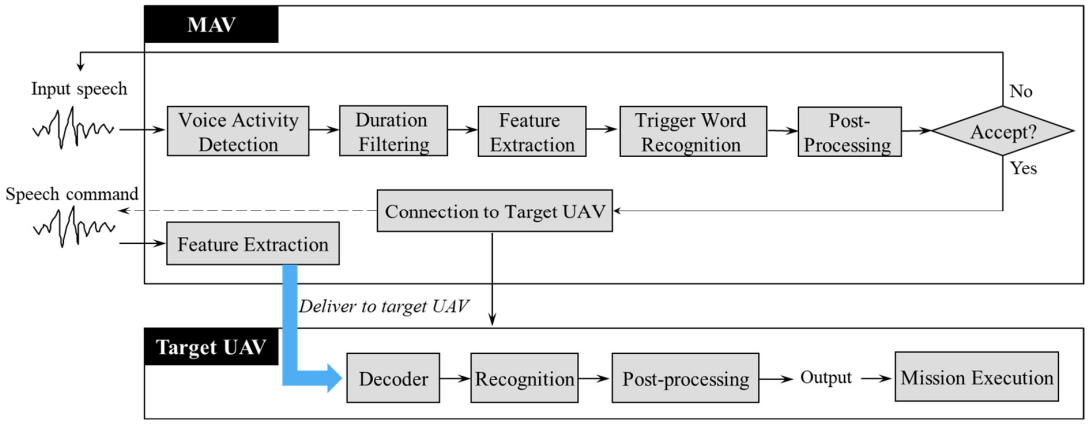

Figure 4. The procedure of multi-channel voice trigger-based front-end and speech recognition.

2.2. Model Construction for Speech Recognition

As described in Figure 4, after the pilot calls the trigger word for the target UAV and a connection is made with the UAV, the pilot delivers a command to the UAV, and the UAV starts speech recognition for this command. This subsection describes the speech recognition process performed in the UAV system.

2.2.1. Definition of Voice Commands for Training Data Collection

For speech recognition, an acoustic model must be constructed in advance (this is called model training), and training data is required in this process. Therefore, for collecting training data, we first define a set of voice commands that the pilot of the MAV delivers to control the UAV. For this work, we conducted expert consultation through several meetings with military aviation officials. The characteristics of commands used for military operations between MAV pilots and UAVs are shown in Table 1.

Table 1. Characteristics of commands used for military operations between MAV pilots and UAVs.

Restrictions	Expert Advice
Structure of commands	Simple and clear commands for precise delivery (1 to 5 connected words)
Vocabulary size	150 to 200 words available to pilots
Language	English is used for communication between military aircraft (International Telecommunications Standard)

In other words, for the voice commands for military operations between MAVs and UAVs, a command structure consisting of 1 to 5 connected words out of approximately 150 to 200 words available to MAV pilots is suitable for accurately delivering commands to the UAV.

The voice command sets must be designed considering various missions the UAV must perform and various collaboration situations between MAV and UAV. In addition, the command sets should be composed of frequently used words for the pilot's convenience and consist of words that are easy for speech recognition. There have been several international cooperation projects related to the collaborative operation of MAVs and UAVs, and various related reports and standards have been published, including the Standardization Agreement (STANAG)-4586, Manned–Unmanned Teaming (MUM-T), and Manned–Unmanned Systems Integration Capability (MUSIC) [18,19]. By analyzing the documents, we investigated the collaboration situations and missions between MAVs and UAVs and specified the division of roles between MAVs and UAVs according to cooperative operation.

In some documents, the core missions of UAVs in the cooperative operation of a MAV and an UAV are known as missions related to reconnaissance, attack, condition monitoring, and location/route management [20,21]. In addition, STANAG-4856, a military standard established by the North Atlantic Treaty Organization (NATO), defines data link interface (DLI) messages between a ground control center and an UAV [18]. Therefore, we derive the voice command sets by linking the UAV core missions with the DLI messages provided by STANAG-4856. Table 2 summarizes the mission command sets configured for several DLI messages.

Table 2. Examples of mission command sets.

DLI Message	Mission Commands
Vehicle Configuration	Check energy storage unit, read back, report energy state, report fuel state, report battery state, ready for launch, acknowledge, are you ready, take off
Vehicle Operating Mode	Set up control mode, request manual control, request automatic control, report control mode
Vehicle Steering	Set up heading point, heading for waypoint (no.), change heading point, report heading point, say heading point, set up altitude, request altitude (no.), maintain altitude, change altitude (no.), say altitude, report altitude, set up speed, reduce speed to (no.), set up loiter position, request loiter position latitude (no.)
Mission Transfer	Set up mission plan, clear route, change route (no.), request route (no.), clear mission, request mission (no.)
AV Loiter Waypoint	Set up loiter type, request loiter type circle, request loiter type racetrack, request loiter radius (no.), report loiter type, report loiter altitude, report loiter speed, request loiter speed (no.), request loiter duration (no.), report loiter duration, request loiter bearing north

Table 3 introduces several scenarios where the UAV executes its mission by passing commands from the MAV pilot to the UAV using the defined command sets. In other words, the phrases in the command scenarios are all examples of commands delivered by the MAV pilot to the UAV, and commands are delivered sequentially according to the phrases in the scenario presented for each mission. It shows that three UAVs (each UAV is named Alpha, Bravo, and Charlie) perform surveillance and reconnaissance under the control of a manned pilot. In a situation where three UAVs are launched simultaneously after the pilot determines basic settings such as route and altitude, Alpha is put into a reconnaissance mission, and Bravo and Charlie are put into surveillance missions. Each mission command starts with calling the UAV to be controlled.

Table 3. Examples of several mission types and command scenarios.

Mission Type	Command Scenario	Mission Type	Command Scenario
Take-off	Agent Alpha Agent Bravo Agent Charlie Set up heading point Heading for waypoint 7 Set up altitude Request altitude 7000 Set up speed Request speed 250 Ready for launch Are you ready Take off Disconnection	Reconnaissance flight instructions	Agent Alpha Request approach Set up altitude Request altitude 3000 Set up area Request vertices number 1 Request area min altitude 2000 Request area max altitude 3000 Request area loop count 10 Disconnection

Table 3. *Cont.*

Mission Type	Command Scenario	Mission Type	Command Scenario
Surveillance flight instructions	Agent Bravo Request activity surveillance Request loiter type circle Request loiter radius 200 Request loiter speed 10 Disconnection Agent Charlie Request activity surveillance Request loiter type figure eight Request loiter speed 20 Disconnection	Return after completing the mission	Agent Alpha Agent Bravo Agent Charlie Clear mission Request flight Change heading point Heading for waypoint 0 Start flight termination Set up control mode Request automatic control Report arrival time Disconnection

2.2.2. Acoustic Model Construction

An acoustic model is a fundamental component of speech recognition [22]. Acoustic model construction is a process of learning to map acoustic features extracted from input speech signals to phonetic units [23,24]. Typical acoustic models are HMM and DNN. HMM is constructed by learning the statistical relationships between the acoustic features and the corresponding phonetic units [25,26]. On the other hand, the DNN-based acoustic model is trained using deep learning techniques to learn a non-linear mapping between the acoustic features and the phonetic units [27].

The HMM is a traditional acoustic model that has been successfully used in many speech recognition systems [24]. It has a simple and interpretable structure and is, therefore, computationally efficient, especially during decoding. However, the HMM has limited ability to model complex non-linear relationships between input features and output phonemes, making it difficult to capture acoustic details [28]. Because of this, the HMM has limitations in recognizing speech with complex structures (e.g., sentence units).

On the other hand, the DNN is capable of modeling complex non-linear relationships between input features and output phonemes, making it possible to capture acoustic details and improve recognition accuracy for sentence-level speech [29,30]. However, it has a more complex structure than HMM, making it more computationally intensive during training and decoding, and it may require specialized hardware to achieve real-time performance. In particular, the DNN requires significant training data to learn many model parameters [31].

Since the HMM and DNN have such conflicting characteristics, selecting and constructing a model suitable for recognizing the mission command sets delivered to the UAV by the MAV pilot and suitable for the system environment driving speech recognition is necessary. As described in the previous section, the pilot's mission commands given to the UAV are relatively short sentences consisting of at most five words. The sentence is considered a series of connected words. The total number of words included in the command sets is only about 400. In terms of the system environment driving speech recognition, UAVs responsible for speech recognition processing typically have limited computing hardware capacity and can perform limited computations.

Based on these characteristics, a speech recognition system that can recognize connected words consisting of a relatively small number of words with medium hardware capacity is appropriate for recognizing UAV mission commands. Therefore, HMM is expected to solve these limitations more effectively than DNN.

For HMM-based connected word recognition, the HMM must be constructed for each word included in the command set. Since the recorded training data are composed of commands in sentence units, it is necessary to segment each speech data sequence into individual words. For each word, a separate HMM is then trained using the segmented

speech data. That is, the same number of HMMs as the number of words included in the command set is constructed during training.

After building an HMM for each word, connected word recognition proceeds as follows. First, by detecting the silence regions included in the input speech, the connected word command is divided into sequences of isolated words. Acoustic features are then extracted from each isolated word. Next, as shown in Figure 5, the same decoding process as isolated word recognition is performed using the Viterbi algorithm [32], which computes the likelihood for each HMM ($\lambda_1, \ldots, \lambda_V$) with given acoustic features. Once an HMM representing the maximum likelihood is determined, the word corresponding to the model is regarded as the recognition result.

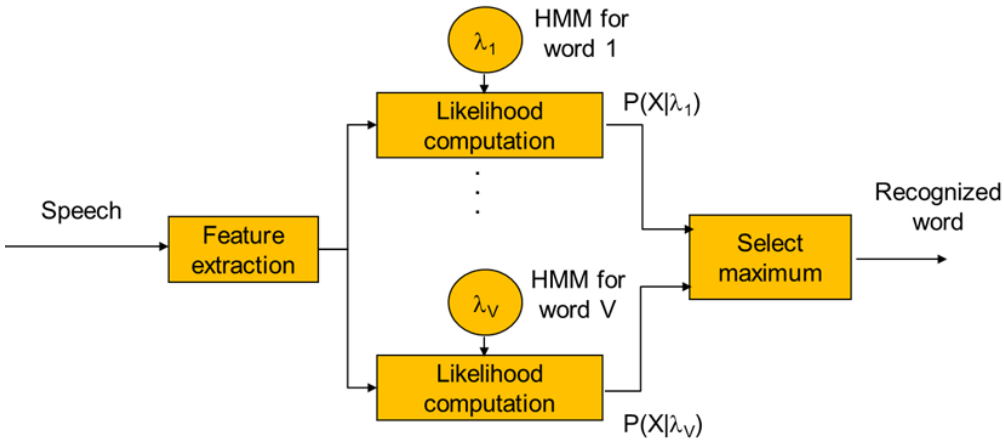

Figure 5. The procedure of HMM-based isolated word recognition.

In the speech recognition process shown in Figure 5, the HMM with the same structure can be used for all words. However, since the amount of information that the model needs to learn varies depending on the length of the word utterance, more effective speech recognition can be performed by modifying the structure of the HMM considering the word length.

As illustrated in Figure 6, if all words have the same HMM structure, inefficient HMMs may be constructed in which one state covers multiple phonemes or several states simultaneously handle one phoneme. In this study, we construct HMMs with different structures for each word to improve the structural problem of HMM. Since we expect that the HMM structure in which one state handles one phoneme is the most effective, we adjust the number of HMM states according to the number of phonemes in each word, as shown in Figure 7.

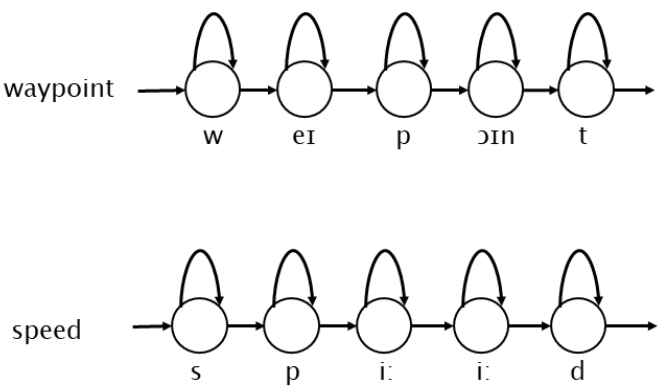

Figure 6. HMM structure with a fixed number of states.

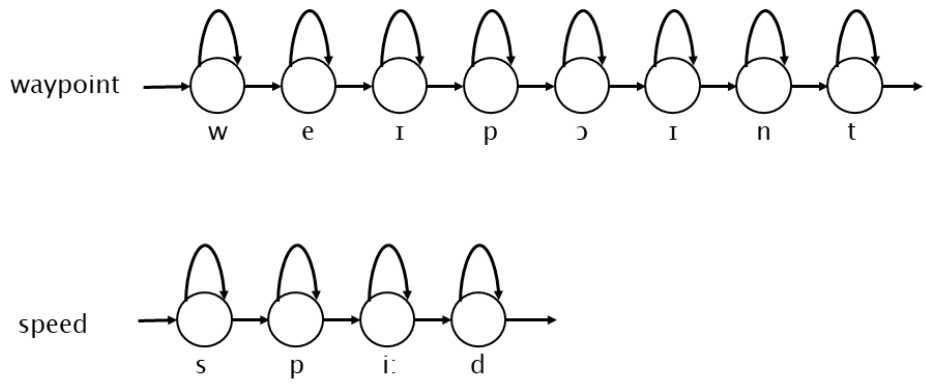

Figure 7. Variable HMM structure considering the number of phonemes in a word.

2.3. Post-Processing with Syntax Analysis and Semantic Analysis

The purpose of post-processing is to improve the accuracy of recognition output through syntax analysis and semantic analysis. In this study, we propose efficient methods for each post-processing task, considering the mission commands established in the collaborative environment of MAVs and UAVs.

2.3.1. Syntax Analysis Based on the Grammar Network

Syntax analysis is the process of analyzing the grammatical structure of a spoken sentence to determine its meaning [33]. It helps clarify sentences with multiple possible meanings. Therefore, this processing is very important in the collaborative environment of MAVs and UAVs where there is a high possibility of misrecognition due to aircraft noise, and the UAV must clarify the pilot's mission commands.

In this study, we organized the grammar structure of mission commands into a tree-type grammar network and performed syntax analysis using this network. That is, the recognition result conforming to this network was determined to have an appropriate grammatical structure; otherwise, it was regarded as misrecognition.

The grammar network has one root node and one terminal node, and the network is formed between the root and the terminal node, with the preceding word becoming the parent node and the succeeding word becoming the child node according to the grammar structure of each command. The starting word of each command becomes the child node of the root node, and the command's last word becomes the terminal node's parent node. When creating a grammar network in this way, several sample commands such as "report loiter altitude", "report loiter duration", and "report battery capacity" can be expressed as a tree, as shown in Figure 8.

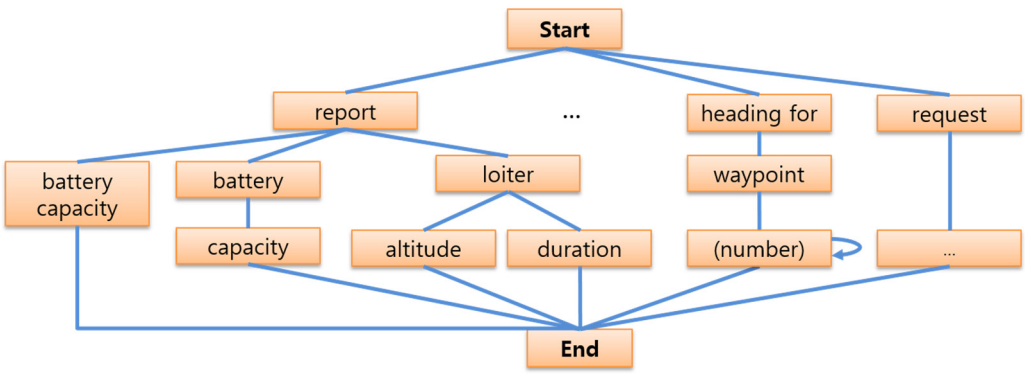

Figure 8. A grammar network created using several sample commands.

In Figure 8, "battery capacity" is handled in two ways: storing two words together in one node or dividing each word into two nodes. The reason for this processing is to recognize it as one word when uttering this command without a pause between two

connected words. In addition, some mission commands contain numbers, such as "heading for waypoint 5". Since numbers have various lengths, they are expressed as one node when processing the syntax of such commands.

The principle that the grammar network we built can be used to determine whether the recognition result conforms to the command syntax is as follows. When a sequence of words included in the recognition result has a path starting from the root node and arriving at a terminal node, the word sequence is determined to conform to the command syntax. If the sequence of words starts from the root node and does not reach the terminal node, the word sequence may not conform to the command syntax.

If it is determined that the word sequence of the recognition result does not conform to the command syntax, the result may be regarded as completely incorrect. Nevertheless, there is a possibility that only one or two words in the word sequence might be incorrectly recognized. Therefore, rather than concluding that the recognition result is completely misrecognized, correcting the misrecognized words may help improve overall performance. In this study, we propose a method to correct such misrecognition of several words by combining candidate recognition results with a grammar network.

In general, when word recognition is performed on connected words of an input command, the similarity between given acoustic features and each word model is calculated. Then, the top several word models selected in order of similarity become candidate recognition results for the given features. At this time, the similarity of each candidate's result is also stored.

Figure 9 shows an example of candidate recognition results for the input command "report loiter altitude". If only the first-rank result of each word is accepted, "report route altitude" becomes the recognition result, which cannot pass through our grammar network. However, as shown in this figure, if the second-rank result of each word is also accepted, "report loiter altitude" can be obtained as a recognition result. In other words, after candidate results for each word are selected, among all possible word sequence combinations constructed from the candidates, a word sequence having the highest similarity while passing through the grammar network becomes the final recognition result of the input command. The grammar network is used in this process to verify whether each word sequence constructed from the candidates matches well with the command syntax.

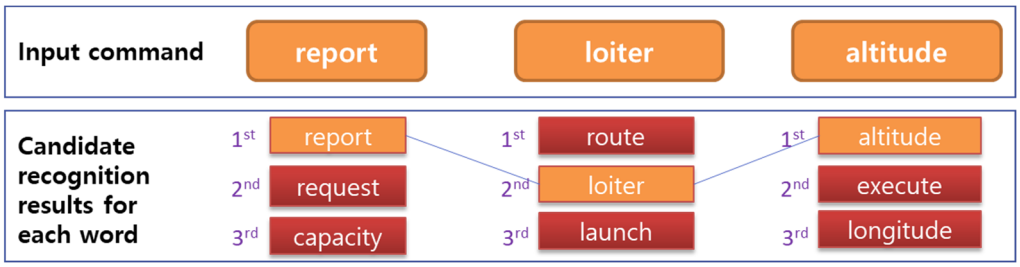

Figure 9. Example of candidate recognition results for the input command "report loiter altitude" and correction of the misrecognition result based on word sequence matching using a grammar network.

As the number of candidate results for each word increases, the amount of computation also increases, so we set the number of candidate results to three, which is considered the most appropriate. If none of all possible word sequence combinations constructed from the candidate results pass through the grammar network, the given input command's recognition result is considered entirely incorrect.

If the first-ranked word sequence does not match the grammar network, it is combined with the next ranked results and attempts to match the grammar network again. If the command consists of five words, the total number of combinations will be $243 = 3^5$. There are rarely situations where all 243 combinations are considered, because usually the first or second ranked results match the grammar network and the matching process stops.

2.3.2. Semantic Analysis Based on Transaction Scheme

Semantic analysis is the process of analyzing the meaning of words and phrases contained in recognition results to extract the intended semantic content [34]. This involves understanding the context of recognition results. It is an important process in speech recognition because it allows systems to accurately transcribe speech into its intended meaning rather than simply recognizing sounds or phonemes. As such, this process is particularly important in applications such as natural language processing and virtual assistants, where understanding the meaning of spoken language is essential for providing accurate and useful responses.

Semantic analysis plays a key role in a system that recognizes mission commands in the collaborative environment of MAVs and UAVs. During an important military operation, if a command transmitted by a pilot is recognized as a command that contradicts the current state of the UAV due to a recognition error or a pilot's ignition mistake, it can encounter a very dangerous situation.

For example, in a situation where the UAV receives the command "Request automatic control", meaning to switch from manual flight to automatic flight and performs the mission, if "Ready for launch" to prepare for take-off is recognized as the next command, the UAV should consider it as a recognition error or a pilot's ignition mistake and report it as an unacceptable command. In this study, we utilize semantic analysis to block dangerous situations caused by recognition errors or pilot ignition mistakes.

The proposed method implements semantic analysis using the transaction scheme used in data management. A transaction refers to a sequence of operations in data management that are treated as a single task unit [35]. It is used to ensure data consistency and integrity through key properties referred to as ACID, which represent atomicity, consistency, isolation, and durability.

Atomicity means that a transaction must be treated as a single indivisible operation, and all operations must succeed or fail as a unit. In other words, if any part of a transaction fails, the entire transaction is rolled back to its previous state. Consistency is the property that the data must be in a consistent state before and after a transaction is executed. Isolation means that the transaction must be executed in isolation from other concurrent transactions. That is, the results of one transaction should not be visible to other transactions until it is committed. Finally, durability is the property that once a transaction is committed, its effect on the data must be permanent. The system can provide reliable and robust data management by ensuring that transactions are ACID-compliant, especially in mission-critical applications where data consistency and integrity are essential.

Because of the characteristics of the transaction, transaction-based semantic analysis of speech recognition results can be effectively used to control UAVs performing critical missions. The process of transaction-based semantic analysis is as follows.

First, critical command sets are defined, such as safety-critical commands, mission-critical commands, and flight-critical commands; then, commands corresponding to each set are selected. Next, each mission is classified as a transaction type, such as a take-off transaction, landing transaction, or reconnaissance transaction. Furthermore, as shown in Figure 10, each critical command set is mapped to a mission transaction, allowing more than one command set to be mapped to a single transaction. Although illustrated here, the mapping information between the critical command set and the mission transaction is managed as a kind of mapping table.

Figure 11 shows the process of making a final decision on whether to accept or reject the command recognition result based on the status of the transaction. We apply the concept of transaction status, commonly used in data management, to this study. A transaction has five statuses: active, partially committed, committed, failed, and aborted. When a transaction starts, it becomes "active". When the transaction ends, it becomes "partially committed", and when it is completely finished, it becomes "committed". On the other hand, if the transaction fails to complete in the active status, it goes into the "failed" status

and eventually changes to the "aborted" status. Sometimes, it is partially committed and becomes a failed status.

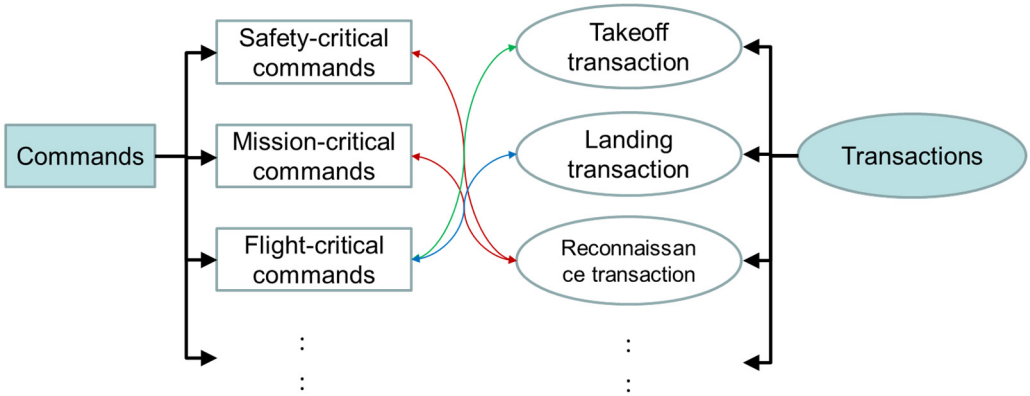

Figure 10. Definition of critical command sets and transaction types and their mapping.

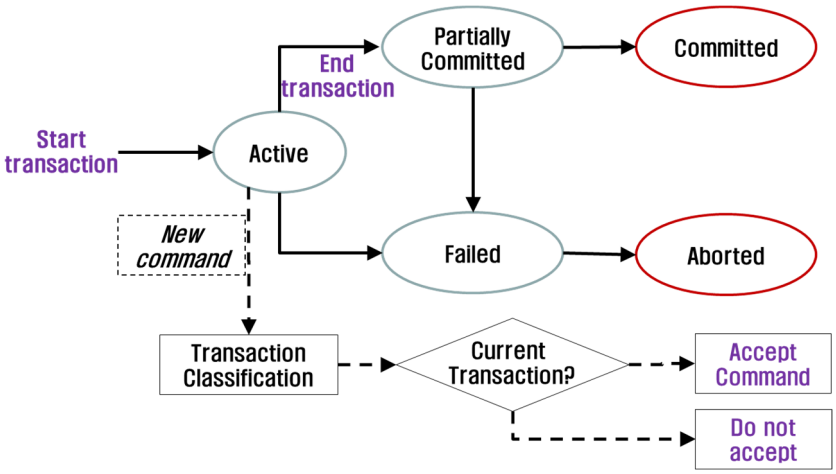

Figure 11. The process of deciding whether to accept the command recognition result based on the transaction status.

The proposed method determines whether to accept or reject the command recognition result based on the transaction status. Assume that a UAV starts a transaction, and a new command is recognized while this transaction is active. At this time, a transaction that the new command corresponds to is found in the mapping table. The new command is accepted if it corresponds to the currently executing transaction; otherwise, it is rejected. As a result, this verification prevents the start of another new transaction before the currently executing transaction becomes committed or aborted.

The proposed method guarantees the independence of individual missions the UAV performs using the transaction scheme. The process for validating recognition results utilizes the ACID characteristics of the transaction discussed above to ensure that the UAV can safely carry out its mission.

Figure 12 shows an example of securing the mission's independence via a UAV by rejecting a disallowed command through transaction-based semantic analysis. In this figure, when a MAV pilot delivers the command "Request activity surveillance" related to the reconnaissance mission to the UAV, the UAV starts the reconnaissance transaction. While performing the reconnaissance transaction, if the UAV recognizes another voice message from the pilot as "Ready for launch" that is related to the take-off mission, the UAV understands that this command is not related to the reconnaissance transaction and informs the MAV that the command is not allowed.

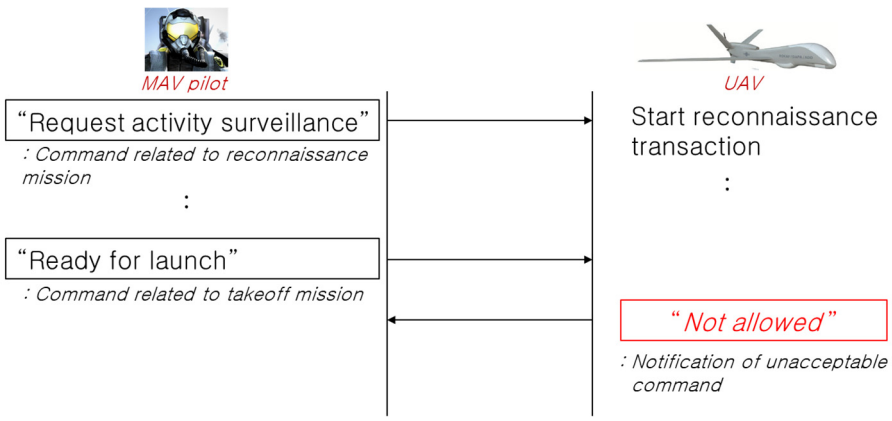

Figure 12. Example of a mission control situation through transaction-based semantic analysis.

3. Evaluation

To validate the efficiency of the proposed speech recognition framework, we conducted several experiments, including speech recognition, syntax analysis, and semantic analysis.

3.1. Validation of Speech Recognition for Mission Command Set in a Collaborative Environment of MAVs and UAVs

Speech recognition experiments were performed to verify the performance of the acoustic models introduced in Section 2.2.2. Training data are required to build acoustic models. As described in Section 2.2.1, we constructed voice command sets related to voice-driven UAV control in a collaborative environment of MAV and UAV, and as a result, obtained about 300 different commands and about 400 different words. We then recorded 50 speakers pronouncing each command and word three times in a clean environment. As a result, 105,000 pieces of voice data (45,000 data for command units and 60,000 for word units) were collected. These data were divided into 10 groups, and speech recognition experiments were conducted using a 10-fold cross-validation method. That is, the data of 5 speakers in the first group were used for testing, and the data of the remaining 45 speakers were used for model training. In this way, the experiments were conducted 10 times by changing the test and training data groups, and the average of each experiment result was calculated.

As explained in Section 2.2.2, we considered the HMM a more efficient model than the DNN in recognizing mission commands in the form of connected words composed of a small number of words. Therefore, an HMM model was built for each word, and connected word recognition experiments were conducted using this model.

In addition, a DNN model was also constructed to compare performance with HMM. However, there is a limit to building a DNN model with about 100,000 voice data we collected, so we built a model using the DARPA Resource Management (RM) speech corpus [36,37]. The DNN used in the experiment is a model with a five-layer structure built based on Kaldi. Kaldi is an automatic speech recognition (ASR) toolkit with many ASR algorithms [38]. It has been released in various versions and has provided various training recipes such as the Wall Street Journal Corpus (wsj), TIMIT (timit), and Resource Management (rm). Since the speech recognition target covered in this study is commands composed of several word sequences, we tried to use a DNN model that shows stable performance while minimizing the amount of computation compared to complex DNN models. For this reason, we trained a TDNN-based triphone model using the Kaldi S5 version by following the recipe using the RM corpus [39,40]. Speech recognition in the two models was performed on a laptop with relatively low specifications (Intel i5 (quad-core, 3.4 GHz), 4 GB RAM) considering the UAV system environment, and the average recognition time was also investigated along with the recognition rate.

Table 4 shows the results. In this experiment, we investigated the command recognition results of sentence units, word error rate, and average recognition time of commands.

In Section 2.2.2, we proposed a method to change the structure of HMM so that each state of HMM processes one phoneme. To verify this method's validity, the performance of the fixed HMM, which consists of a fixed number of states for all words, and the variable HMM, which has a different number of states for each word, were compared. In order to examine the results more elaborately, recognition experiments were conducted according to the length of the command, from a command consisting of two words to a command consisting of five or more words.

Table 4. Performance of acoustic models (HMM and DNN): recognition rate by sentence (%), word error rate (%), and average recognition time (sec) in command units.

Model	Measure	2 Words	3 Words	4 Words	5 or More
Fixed HMM	Rec. Rate (sent.)	100	97.4	95.3	94.2
	Word Error Rate	0	0.87	1.49	2.15
	Avg. Rec. Time *	0.03	0.05	0.06	0.09
Variable HMM (proposed)	Rec. Rate (sent.)	100	98.5	97.4	96.9
	Word Error Rate	0	0.54	0.86	1.13
	Avg. Rec. Time *	0.03	0.04	0.05	0.10
DNN	Rec. Rate (sent.)	100	98.8	97.6	97.2
	Word Error Rate	0	0.49	0.80	1.09
	Avg. Rec. Time *	2.0	4.5	6.5	9.0
	Avg. Rec. Time **	0.20	0.38	0.55	0.85

* Experiment with a low-spec laptop/** Experiment with a high-spec laptop.

As shown in this table, the proposed variable HMM improved performance compared to the fixed HMM. The longer the command length, the more noticeable the performance improvement. However, there was no significant difference in recognition time between the two models. Therefore, this result indicates that the variable HMM configures the number of states differently for each word and is more efficient in recognizing connected words.

Next, we compared the performance of the proposed variable HMM and DNN models. In the experimental results, there was not much difference between the two models in the recognition accuracy, while showing a good performance of 97% or more. As a result of recognizing whole command units consisting of two to five words, the proposed variable HMM and DNN showed an average recognition rate of 98.2% and 98.4%, respectively, derived from four types of sentence recognition rates (ranging from two words to five or more words). For commands composed of two words, both models showed 100% accuracy, and for other commands, the performance of HMM was slightly lower than that of DNN. Among the two measures of recognition accuracy, the word error rate showed a much smaller difference between the two models, and for commands consisting of five or more words, the performance difference was only 0.04%. The reason is that recognition errors in sentence units are mostly caused by the misrecognition of only one word included in a sentence.

On the other hand, the two models showed a big difference in average recognition time. The HMM showed a recognition time shorter than 0.1 s for command sets of all lengths, while the DNN showed a significantly longer recognition time as the command length increased, ranging from 2 s (two words) to 9 s (five or more words). This result is because the DNN has a more complex model structure than the HMM.

Considering that the non-ideally high recognition time of the DNN model was due to the influence of the laptop used in the experiment, we additionally measured the recognition time of the DNN model using a high-spec laptop (Intel i7 (12-core, 2.1 GHz), 16 GB RAM). This result is shown in the last row of Table 4. Because the same program was evaluated on both laptops, the recognition results did not change, but the average recognition time showed a difference. For commands of each length, the second laptop showed recognition times ranging from 0.2 to 0.9 s, reducing the average recognition time by about 10 times compared to the results in the first laptop. However, even with high-spec

hardware, the HMM model still showed about 10 times lower recognition time. Therefore, it can be said that HMM, which has a relatively simple structure, is a suitable model to recognize the UAV control commands targeted in this study.

We measured the amount of computation using the number of model parameters to examine the theoretical difference in recognition time between the HMM model and the DNN model. The amount of computation required to process one speech frame in the DNN model can be calculated through the following equation.

$$N^{(DNN)} \approx (L-1) * N^2 + N * D + N * S, \tag{1}$$

where L is the number of layers, N is the number of nodes in each layer, D is the dimension of the feature vector, and S represents the number of nodes in the output layer. The values of the Kaldi model parameter used in our experiment are as follows: L = 5, N = 650, D = 39, and S = 2000. Applying these values to (1), about 3 million operations are required to process one frame during the recognition process.

On the other hand, in the case of the HMM model, the amount of computation required to process one speech frame is calculated as follows.

$$N^{(HMM)} \approx S * M * D * 2, \tag{2}$$

where S is the number of HMM states, M is the number of GMM mixtures, and D represents the dimension of the feature vector. The values of the HMM model parameters used in our experiment are as follows: S = 100, M = 8, and D = 39. Accordingly, about 30,000 operations are required to process one frame during the recognition process.

That is, the DNN model requires about 100 times the amount of computation of the HMM model. Furthermore, this theoretical difference is similarly shown in Table 4, with the average recognition time of the DNN model showing a value about 100 times higher than that of the HMM model.

3.2. Verification of the Proposed Syntax Analysis Method for the Post-Processing of Mission Command Speech Recognition

We verified the validity of the proposed grammar network-based syntax analysis method through speech recognition experiments. As explained in Figure 9, only the first-rank recognition result of each word is accepted in a general speech recognition framework. The speech recognition results in Table 4 are the recognition accuracy considering only the first-rank results.

However, as mentioned in Section 2.3.1, when the correct answer of a specific word among connected words is ranked second or third, the recognition accuracy can be increased if these candidates are also considered. To implement this, we proposed a syntax analysis method using a grammar network. If the first-rank recognition result is found to be incorrect in the command syntax through the grammar network, the optimal result that matches the command syntax is obtained by combining the second or third-rank candidate results.

Table 5 represents the speech recognition results after applying the proposed syntax analysis method. The proposed variable HMM, which was determined to be the most efficient model in terms of recognition rate and average recognition time in Table 4, was set as the baseline. Furthermore, the recognition rate was investigated after performing grammar network-based syntax analysis for the three best candidate results (that is, the recognition results up to the third rank in the order of output values calculated in HMMs).

Since the commands composed of two words had all correct answers at the first rank in the baseline, the combination of the first-rank words conformed to the command syntax, and therefore, all were recognized as correct answers after applying the syntax analysis. In the case of commands composed of three or more words, the recognition rate increased in all lengths of command sets after applying the proposed syntax analysis method. These results demonstrate that when some of the connected words constituting a command are incorrect, the baseline treats the command as an error, but in the proposed method, many

of these data are corrected as the correct answers. In particular, the longer the length of the command, the higher the improvement in the recognition rate by the syntax analysis, which explains that the longer the length of the command, the more errors occur, and the proposed method corrects the errors.

Table 5. Speech recognition results (%) of the whole command unit (not word unit) for validation of the proposed grammar network-based syntax analysis method.

	2 Words	3 Words	4 Words	5 or More
Baseline	100	98.5	97.4	96.9
Applying syntax analysis (proposed)	100	99.7	99.0	98.8

3.3. Verification of the Proposed Semantic Analysis Method for the Post-Processing of Mission Command Speech Recognition

The evaluation of speech recognition models or syntax analysis can be quantitatively evaluated through speech recognition experiments, but the quantitative method is not suitable for evaluating semantic analysis. Accordingly, we implemented a real-time recognition system and a collaborative simulation environment of MAVs and UAVs and attempted to verify the validity of the proposed semantic analysis method.

Figure 13a shows a program for real-time command recognition, and Figure 13b represents an experimental environment created to simulate the collaboration of three UAVs named Alpha, Bravo, and Charlie with a MAV pilot. In this simulation environment, when the pilot issues a command to a device indicating the MAV, the device transmits the command to the target UAV, and then, the UAV recognizes the command. All these processes run in real-time.

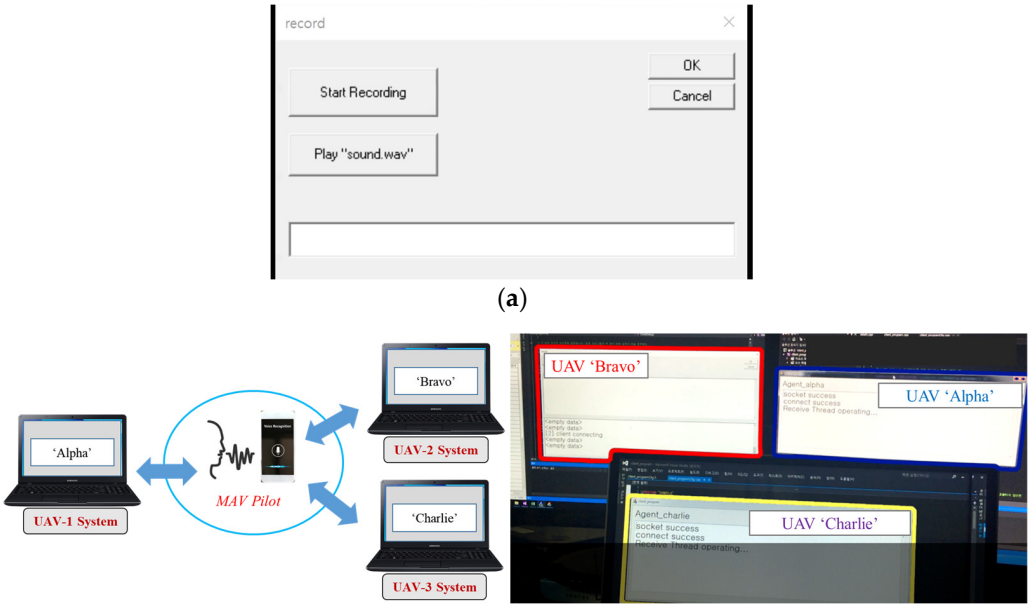

Figure 13. Program for real-time command recognition (**a**) and experimental environment for simulating the collaboration of MAVs and multi-UAVs (**b**).

We created several collaboration scenarios between MAVs and UAVs in this simulation environment to verify whether the proposed transaction-based semantic analysis method works properly while MAVs and UAVs communicate. Figure 14 shows flow charts configured for collaboration scenarios between a MAV and an UAV. The two figures represent collaboration examples: (a) is a collaboration scenario related to condition monitoring and location/route management, and (b) is a scenario representing reconnaissance and

attack. We made several such scenarios and tested whether the transaction-based semantic analysis works properly while communicating between the MAV device and the UAV system in real-time in the MAV and UAV collaborative simulation environment presented in Figure 13b.

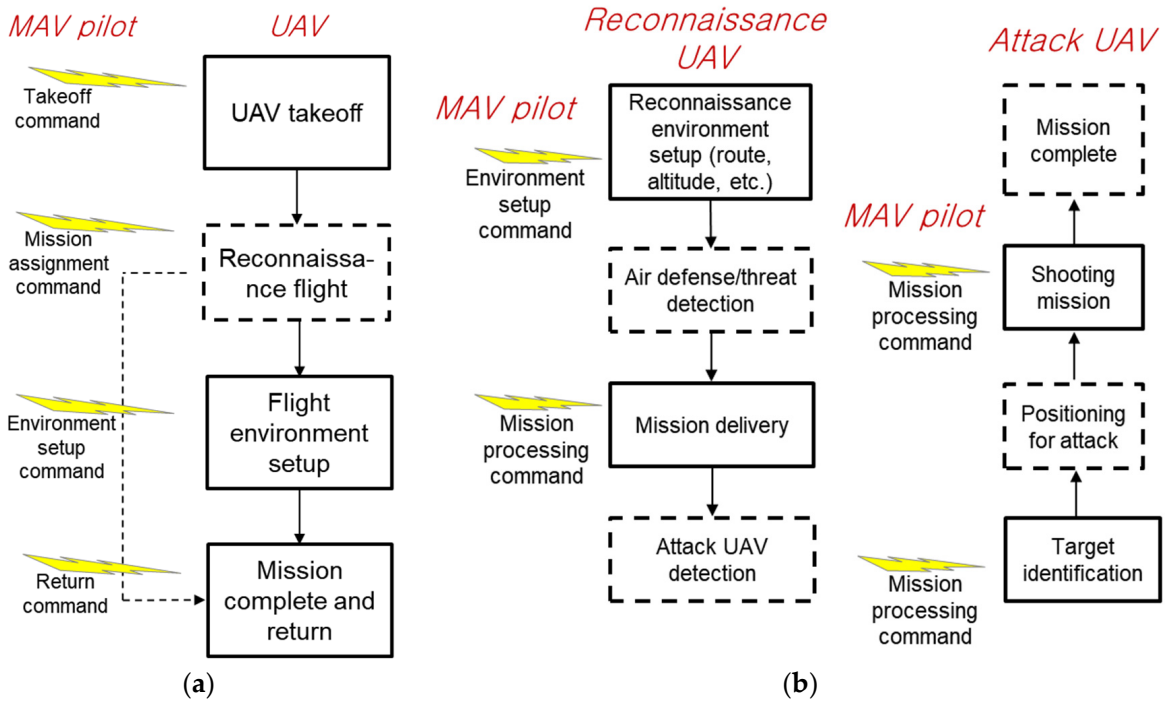

Figure 14. Flow chart for collaboration scenarios between MAVs and UAVs. (**a**) shows the takeoff scenario, and (**b**) illustrates the reconnaissance and attack scenario.

In Figure 14, the arrows indicate the transaction flow, and the yellow signs indicate the situation where the MAV pilot delivers a command to the UAV. The solid boxes represent the transactions the UAV is performing, and the dotted boxes represent the status of the UAV or the situation the UAV is in. For example, in Figure 14a, the pilot issues a take-off command, and the UAV enters a transaction called take-off. Afterward, when the pilot issues a command related to reconnaissance flight, the UAV enters a reconnaissance flight state. Then, when the pilot issues an environment setup command, the UAV enters the Flight environment setup transaction. Finally, when the return command is delivered to the UAV, it changes the UAV into a transaction called mission complete and return. It is impossible for a transaction to start in an order different from the direction of the transactions indicated by the arrow in the figure. For example, after the flight environment setup transaction, the UAV can only enter the mission complete and return transaction and cannot proceed to the take-off transaction. With this process based on a transaction concept, we implement flow control of UAVs collaborating with MAVs and utilize this for semantic analysis of commands to prevent serious situations caused by incorrect speech recognition results.

We conducted an experiment to verify that the proposed transaction-based semantic analysis works properly. For example, in the scenario shown in Figure 14a, let us consider a situation in which a command related to a take-off transaction is entered while the MAV pilot delivers the environment setup command and the UAV performs a transaction of flight environment setup. At this time, we checked whether the UAV system correctly rejected this command by considering it as a context error. As another example, in the attack scenario of Figure 14b, when a command related to the landing transaction was entered while performing a transaction of shooting mission, we checked whether the UAV rejected this command correctly.

This way, we checked whether 100 normal commands that matched any given transaction were correctly accepted and 100 abnormal commands that violated the transaction were accurately rejected. As a result of the experiment, it was found that all normal commands were correctly accepted, and all abnormal commands were rejected by the proposed transaction-based semantic analysis method.

4. Conclusions

In this study, we proposed a speech recognition framework for voice-driven UAV control in a collaborative environment of MAVs and UAVs. The previous study proposed an efficient noise-cancellation method in an aerial vehicle environment and a multi-channel voice-triggering method for controlling multiple UAVs for front-end speech recognition. In this study, we focused on constructing acoustic models for speech recognition and post-processing to perform syntax analysis and semantic analysis. In a collaborative environment between MAVs and UAVs, typical commands that a MAV pilot sends to UAVs are in the form of connected words consisting of at most five words. This study investigated model construction and post-processing methods suitable for recognizing such connected words in a UAV system with low hardware capacity.

First, we explored an efficient acoustic model for recognizing connected words, targeting the HMM, known as the statistical modeling method, and the DNN model using deep learning techniques. In particular, instead of the traditional method using the same HMM structure for each word, we proposed a HMM structure that reflects the number of phonemes in a word. In the average recognition rate of four types of sentence recognition rates (from commands consisting of two words to commands of five or more words), the DNN-based acoustic model showed higher performance than the traditional HMM, while it did not show much difference from the proposed HMM. However, in terms of the amount of computation and recognition time, it was analyzed that the HMM model performs fast recognition with about 100 times less computation than the DNN model. Furthermore, it can be concluded that the proposed HMM model is suitable for recognizing connected words in a UAV system with low hardware capacity.

Naturally, among the various DNN models currently in use, there are models with relatively low computational complexity. Although this study highlighted the fact that the HMM model is less computationally intensive than DNN, this is not a claim that the HMM is the optimal model, and it can be used as an alternative in constrained environments. If the UAV system has high-performance hardware capacity and can allocate many resources to driving speech recognition, the DNN model will also be available.

Next, a grammar network-based syntax analysis method was proposed for post-processing. We configured the structure of the commands that the MAV pilot delivers to the UAV as a grammar network, and if the connected words obtained as a recognition result do not pass through this network, it is determined as a syntax error. In addition, when some of the connected words contain errors, instead of treating the corresponding command as an error, we corrected the recognition error by reflecting the results in the upper rank among the candidate results of each word using the proposed syntax analysis method. As a result of the speech recognition experiment conducted to verify the validity of this method, it was confirmed that the speech recognition performance was remarkably improved after applying the proposed syntax analysis method. As a result of recognizing whole command units consisting of two to five words, the average recognition rates of the baseline approach and the syntax analysis-based approach were 98.2% and 99.4%, respectively, which means that the relative improvement in error rate by the syntax analysis reaches 65%.

Finally, we proposed a semantic analysis approach applying the transaction scheme used in data management. In a very important situation, such as a military operation, misrecognition of a MAV pilot's command may lead to serious danger. To handle this situation, we categorized cooperation missions between MAVs and UAVs as transactions and mapped each command set to related transactions. Then, while the UAV is performing a transaction corresponding to a specific mission when the recognition result of a command

delivered by the MAV pilot does not belong to the transaction, the UAV regards it as a recognition error and sends a response indicating that the command cannot be accepted.

To verify the validity of this method, we implemented a real-time recognition system and a collaborative simulation environment of MAVs and UAVs and created several collaboration scenarios between a MAV and an UAV. Then, real-time communication between the MAV and the UAV was performed using the scenarios to confirm that the proposed semantic analysis works properly. In experiments conducted with about 200 commands, it was confirmed that normal commands that match a given transaction are correctly accepted, and commands that do not match are properly rejected.

As described so far, in this study, we introduced a speech recognition framework for voice-based UAV control in a collaborative environment between MAVs and UAVs, proposed useful methods in each process, and successfully verified the validity of each module through various speech recognition experiments. The proposed framework consists of speech database construction, front-end, acoustic model construction, and post-processing and focuses on minimizing the amount of computation so that each module can be directly driven in the UAV system. Therefore, the framework is expected to be efficiently applied in an environment where speech recognition is directly driven in a device with limited hardware resources.

In future research, we plan to expand this research by studying an efficient speech recognition framework for voice-driven communication between the ground control center and an UAV and between the ground control center and a MAV.

Author Contributions: Conceptualization, J.-S.P.; methodology, J.-S.P.; software, J.-S.P. and N.G.; validation, J.-S.P. and N.G.; formal analysis, J.-S.P.; writing—original draft preparation, J.-S.P.; writing—review and editing, J.-S.P. and N.G.; supervision, J.-S.P.; project administration, J.-S.P.; funding acquisition, J.-S.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the Hankuk University of Foreign Studies Research Fund, the National Research Foundation of Korea (NRF) grant funded by the Korean Government (MSIT) (No. 2020R1A2C1013162), and the Institute of Information and Communications Technology Planning and Evaluation (IITP) grant funded by the Korean Government (MSIT) (RS-2023-00232949).

Data Availability Statement: Data is contained within the article.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Oneata, D.; Cucu, H. Kite: Automatic speech recognition for unmanned aerial vehicles. *arXiv* **2019**, arXiv:1907.01195.
2. Lavrynenko, O.Y.; Konakhovych, G.F.; Bakhtiiarov, D.I. Protected voice control system of unmanned aerial vehicle. *Electr. Control Syst.* **2020**, *1*, 92–98. [CrossRef]
3. Anand, S.S.; Mathiyazaghan, R. Design and fabrication of voice controlled unmanned aerial vehicle. *IAES Int. J. Robot. Autom.* **2016**, *5*, 205–212. [CrossRef]
4. Park, J.S.; Na, H.J. Front-end of vehicle-embedded speech recognition for voice-driven multi-UAVs control. *Appl. Sci.* **2020**, *10*, 6876. [CrossRef]
5. Helmke, H.; Kleinert, M.; Shetty, S.; Ohneiser, O.; Ehr, H.; Arilíusson, H.; Simiganoschi, T.S.; Prasad, A.; Motlicek, P.; Veselý, K.; et al. Readback error detection by automatic speech recognition to increase ATM safety. In Proceedings of the Fourteenth USA/Europe Air Traffic Management Research and Development Seminar (ATM2021), Virtual Event, 20–23 September 2021; pp. 20–23.
6. Helmke, H.; Kleinert, M.; Ahrenhold, N.; Ehr, H.; Mühlhausen, T.; Ohneiser, O.; Klamert, L.; Motlicek, P.; Prasad, A.; Zuluaga-Gomez, J.; et al. Automatic speech recognition and understanding for radar label maintenance support increases safety and reduces air traffic controllers' workload. In Proceedings of the Fifteenth USA/Europe Air Traffic Management Research and Development Seminar (ATM2023), Savannah, GA, USA, 5–9 June 2023; pp. 1–11.
7. Guo, D.; Zhang, Z.; Fan, P.; Zhang, J.; Yang, B. A context-aware language model to improve the speech recognition in air traffic control. *Aerospace* **2021**, *8*, 348. [CrossRef]
8. Zhang, S.; Kong, J.; Chen, C.; Li, Y.; Liang, H. Speech GAU: A single head attention for Mandarin speech recognition for air traffic control. *Aerospace* **2022**, *9*, 395. [CrossRef]
9. Lin, Y. Spoken instruction understanding in air traffic control: Challenge, technique, and application. *Aerospace* **2021**, *8*, 65. [CrossRef]

10. Oneață, D.; Cucu, H. Multimodal speech recognition for unmanned aerial vehicles. *Comput. Electr. Eng.* **2021**, *90*, 106943. [CrossRef]
11. Xiang, X.; Tan, Q.; Zhou, H.; Tang, D.; Lai, J. Multimodal fusion of voice and gesture data for UAV control. *Drones* **2022**, *6*, 201. [CrossRef]
12. Galangque, C.M.J.; Guirnaldo, S.A. Speech recognition engine using ConvNet for the development of a voice command controller for fixed wing unmanned aerial vehicle (UAV). In Proceedings of the 12th International Conference on Information & Communication Technology and System (ICTS), Surabaya, Indonesia, 18 July 2019; pp. 93–97. [CrossRef]
13. Zhou, Y.; Hou, J.; Gong, Y. Research and application of human-computer interaction technology based on voice control in ground control station of UAV. In Proceedings of the IEEE 6th International Conference on Computer and Communications (ICCC), Chengdu, China, 11–14 December 2020; pp. 1257–1262. [CrossRef]
14. Contreras, R.; Ayala, A.; Cruz, F. Unmanned aerial vehicle control through domain-based automatic speech recognition. *Computers* **2020**, *9*, 75. [CrossRef]
15. Trivedi, A.; Pant, N.; Shah, P.; Sonik, S.; Agrawal, S. Speech to text and text to speech recognition systems-a review. *IOSR J. Comput. Eng.* **2018**, *20*, 36–43.
16. Karpagavalli, S.; Chandra, E. A review on automatic speech recognition architecture and approaches. *Int. J. Signal Process. Image Process. Pattern Recognit.* **2016**, *9*, 393–404. [CrossRef]
17. Desai, N.; Dhameliya, K.; Desai, V. Feature extraction and classification techniques for speech recognition: A review. *Int. J. Emerg. Technol. Adv. Eng.* **2013**, *3*, 367–371.
18. Marques, M.M. STANAG 4586—Standard interfaces of UAV control system (UCS) for NATO UAV interoperability. *NATO Stand. Agency Afeite Port.* **2012**, *3*, 1–14.
19. Kim, S.; Kim, Y. Development of an MUM-T integrated simulation platform. *IEEE Access.* **2023**, *11*, 21519–21533. [CrossRef]
20. Jameson, S.; Franke, J.; Szczerba, R.; Stockdale, S. Collaborative autonomy for manned/unmanned teams. In Proceedings of the Annual Forum American Helicopter Society, Grapevine, TX, USA, 1–3 June 2005; Volume 61, p. 1673.
21. Alicia, T.J.; Hall, B.T.; Terman, M. Synergistic Unmanned Manned Intelligent Teaming (SUMIT). In *Technical Report*; U.S. Army: Madison County, NY, USA, 2020; pp. 1–92.
22. Juang, B.H.; Rabiner, L.R. Hidden Markov models for speech recognition. *Technometrics* **1991**, *33*, 251–272. [CrossRef]
23. Woodland, P.C.; Odell, J.J.; Valtchev, V.; Young, S.J. Large vocabulary continuous speech recognition using HTK. In Proceedings of the ICASSP'94, IEEE International Conference on Acoustics, Speech and Signal Processing, Adelaide, Australia, 19–22 April 1994; Volume 2, pp. II/125–II/128. [CrossRef]
24. Mor, B.; Garhwal, S.; Kumar, A. A systematic review of hidden Markov models and their applications. *Arch. Comput. Methods Eng.* **2021**, *28*, 1429–1448. [CrossRef]
25. Gales, M.; Young, S. The application of hidden Markov models in speech recognition. *Found. Trends Signal Process.* **2007**, *1*, 195–304. [CrossRef]
26. Mustafa, M.K.; Allen, T.; Appiah, K. A comparative review of dynamic neural networks and hidden Markov model methods for mobile on-device speech recognition. *Neural Comput. Appl.* **2019**, *31*, 891–899. [CrossRef]
27. Hinton, G.; Deng, L.; Yu, D.; Dahl, G.E.; Mohamed, A.; Jaitly, N.; Senior, A.; Vanhoucke, V.; Nguyen, P.; Sainath, T.N.; et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Process. Mag.* **2012**, *29*, 82–97. [CrossRef]
28. Shahin, M.A.; Ahmed, B.; McKechnie, J.; Ballard, K.J.; Gutierrez-Osuna, R. A comparison of GMM-HMM and DNN-HMM based pronunciation verification techniques for use in the assessment of childhood apraxia of speech. *Interspeech* **2014**, *1*, 1583–1587.
29. Fohr, D.; Mella, O. New paradigm in speech recognition: Deep neural networks. In Proceedings of the International Conference on Information Systems and Economic Intelligence, Marrakech, Morocco, 13 April 2017.
30. Këpuska, V.; Bohouta, G. Comparing speech recognition systems (Microsoft API, Google API and CMU Sphinx). *Int. J. Eng. Res. Appl.* **2017**, *7*, 20–24. [CrossRef]
31. Deshmukh, A.M. Comparison of hidden Markov model and recurrent neural network in automatic speech recognition. *Eur. J. Eng. Res. Sci.* **2020**, *5*, 958–965. [CrossRef]
32. Lou, H.L. Implementing the Viterbi algorithm. *IEEE Signal Process. Mag.* **1995**, *12*, 42–52. [CrossRef]
33. Arora, S.J.; Singh, R.P. Automatic speech recognition: A review. *Int. J. Comput. Appl.* **2012**, *60*, 34–44. [CrossRef]
34. Tur, G.; DeMori, R. *Spoken Language Understanding: Systems for Extracting Semantic Information from Speech*; John Wiley and Sons: Hoboken, NJ, USA, 2011.
35. Bernstein, P.A.; Newcomer, E. *System Recovery, In Principles of Transaction Processing*; Morgan Kaufmann: San Francisco, CA, USA, 2009; pp. 185–222, ISBN 9781558606234.
36. Hain, T.; Woodland, P.C. Dynamic HMM selection for continuous speech recognition. In Proceedings of the 6th European Conference on Speech Communication and Technology (EUROSPEECH 1999), Budapest, Hungary, 5–9 September 1999.
37. Pallett, D.S.; Fiscus, J.G.; Garofolo, J.S. DARPA resource management benchmark test results June 1990. In Proceedings of the Workshop on Speech and Natural Language, Hidden Valley, PA, USA, 24–27 June 1990.
38. Povey, D.; Ghoshal, A.; Boulianne, G.; Burget, L.; Glembek, O.; Goel, N.; Hannemann, M.; Motlicek, P.; Qian, Y.; Schwarz, P.; et al. The Kaldi speech recognition toolkit. In Proceedings of the IEEE 2011 Workshop on Automatic Speech Recognition and Understanding (ASRU 2011), Waikoloa, HI, USA, 11–15 December 2011.

39. Kaldi Tutorial. Available online: https://kaldi-asr.org/doc/tutorial.html (accessed on 10 January 2023).
40. GitHub: Kaldi Speech Recognition Toolkit. Available online: https://github.com/kaldi-asr/kaldi (accessed on 10 January 2023).