


Article

# Customization of the ASR System for ATC Speech with Improved Fusion

Jiahao Fan  and Weijun Pan \*

College of Air Traffic Management, Civil Aviation Flight University of China, Guanghan 618307, China; 15001077238@163.com

\* Correspondence: wjpan@cafuc.edu.cn

**Abstract:** In recent years, automatic speech recognition (ASR) technology has improved significantly. However, the training process for an ASR model is complex, involving large amounts of data and a large number of algorithms. The task of training a new model for air traffic control (ATC) is considerable, as it may require many researchers for its maintenance and upgrading. In this paper, we developed an improved fusion method that can adapt the language model (LM) in ASR to the domain of air traffic control. Instead of using vocabulary in traditional fusion, this method uses the ATC instructions to improve the LM. The perplexity shows that the LM of the improved fusion is much better than that of the use of vocabulary. With vocabulary fusion, the CER in the ATC corpus decreases from 0.3493 to 0.2876. The improved fusion reduces the CER of the ATC corpora from 0.3493 to 0.2761. Although there is only a difference of less than 2% between the two fusions, the perplexity shows that the LM of the improved fusion is much better.

**Keywords:** speech recognition; transfer learning; air traffic control; language model



**Citation:** Fan, J.; Pan, W. Customization of the ASR System for ATC Speech with Improved Fusion. *Aerospace* **2024**, *11*, 219. <https://doi.org/10.3390/aerospace11030219>

Academic Editor: Álvaro Rodríguez-Sanz

Received: 12 February 2024

Revised: 6 March 2024

Accepted: 10 March 2024

Published: 12 March 2024



**Copyright:** © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

The primary task of air traffic control (ATC) is to prevent aircraft from colliding and to facilitate the smooth and orderly flow of air traffic. In ATC, an air traffic controller (ATCO) directs a flight by giving voice instructions to the pilot. This communication between ATCOs and pilots is called Pilot-Controller Voice Communication (PCVC). In current ATC management systems, PCVC is a concentrated human-in-the-loop (HITL) procedure [1]. The HITL procedure is considered a safety risk. To monitor the HITL risk, ATC management systems need to understand PCVC. The machines can understand real-time ATC voice conversations through text to support further applications. Automatic speech recognition (ASR) is a powerful interface for human-machine interaction that can translate speech into text. ASR is also expected to be a promising bridge between humans (ATCOs and pilots) and the ATC system [2]. Researchers have attempted to improve the accuracy of ASR in the ATC domain. For example, Guo [3] proposed a context-aware language model, and Zhang [4] added a gated attention unit to ASR systems.

Most of the research for ASR in ATC has followed the ASR technology. Most of the most advanced ASR systems currently available consist of three different modules: the acoustic model (AM), the pronunciation model (PM), and the language model (LM) [5]. Deep learning ASR systems have mainly adopted a hybrid approach, where the AM is replaced by a deep neural network while the remaining modules use traditional methods. In the ATC domain, the ATCSpeechNet framework [6,7] had the same structure as AM, PM, and LM. The current trend in the development of ASR systems is towards the creation of end-to-end (E2E) deep neural networks, which can map an input speech sequence directly to a sequence of graphemes, characters, or words. In E2E ASR systems, the acoustic, pronunciation, and language modules are trained together to optimize a unified objective function, thus overcoming the limitations of traditional ASR systems [8–10]. It is the same as Zhang's research [4].

All of the above research has attempted to train new models to perform better in the ATC domain. But more recently, ASR technology has become more accurate and robust. Many ASR services are provided by cloud service providers. For example, Microsoft Azure [11], Amazon Transcribe [12], Huawei Cloud [13], etc. all provide ASR (speech-to-text) services. These big companies can acquire more speech data to improve the accuracy of ASR. In addition, their ASR systems have been tested more often, which makes those ASR systems more robust. In this paper, we believe that ASR in the ATC domain needs to follow transfer learning (TL) technology. This means that researchers should focus on adapting ASR rather than training new ASR models. Using transfer learning to develop an ATC-specific model can significantly reduce the effort required for these systems. ASR services for large companies always have some custom methods. These services can accept vocabularies or audio data with reference transcriptions [14]. The vocabulary can improve the recognition of terminology. The audio data can improve the recognition of a specific audio state.

In this research, we adapted the ASR model using the text data of ATC instructions, which contain more information than vocabulary and are easier to collect than the labeled audio data. To test the adaptation method, we trained a recurrent neural network (RNN)-based ASR model. We compared our method with methods provided by ASR services. The rest of this paper is structured as follows: Section 2 introduces the related work of TL, especially in ASR. TL is the basis of customization. In Section 3, we develop our method and distribute the difference between our method and the traditional method. The test ASR model is also introduced in this section. In Section 4, the ASR model is trained with the Aishell corpus and tested for the performance of different methods on PCVC. Finally, in Sections 5 and 6, we summarize and discuss the main findings and give an outlook on our future work.

## 2. Related Work

Machine learning or deep learning methods work optimally under one fundamental assumption: the training and test data come from an identical feature space and follow the same distribution. A shift in the distribution typically requires most statistical models to be rebuilt from scratch, using freshly collected training data. In many practical applications, retrieving the necessary training data and rebuilding the models prove to be exorbitantly expensive or infeasible [15]. It would be advantageous to reduce the need for and effort associated with training data retrieval. In such cases, TL between task domains would be desirable.

There are many TL techniques for ASR. In this section, we have categorized the TL methods relevant to ASR into those related to AM and LM. Transfer learning for PMs does not exist. However, some ASR services use standardized pronunciations for specific terminology, a practice that is more akin to LM than PM. In this paper, we have focused on LM, as all the methods in this paper are LM-based.

### 2.1. TL of Acoustic Model (AM)

The E2E and layered DNN-HMM frameworks represent two major categories of contemporary deep learning-based AMs. In AM, the DNN is central to the extraction of high-level features from acoustic signals, such as Mel Frequency Cepstral Coefficients (MFCCs), with the subsequent need for HMM lexical sequences for transcription decoding. The DNN processes acoustic attributes as input, yielding context-dependent lexical units, which then serve as input to the downstream HMM component. Conversely, the end-to-end model is a pure DNN approach that takes acoustic features as input and produces the recognition result directly. The neural network derives embeddings from the input features, which are then fed into a series of recurrent layers. These recurrent layers produce a final output by identifying patterns based on both previous and current input data. The network is trained to use backpropagation in conjunction with the CTC (Connectionist Temporal Classification) loss function [16]. Broadly speaking, three primary TL methods

have been used in the context of AMs [5]: feature normalization, discriminative training, and subspace-based TL.

## 2.2. TL of Language Model (LM)

The backoff n-gram model is widely used as a language model (LM) due to its widespread use in automatic speech recognition (ASR) systems. The backoff n-gram LM can be characterized as an aggregation of tuples, each consisting of an n-gram unit and its corresponding logarithmic probability value [17]. This transfer learning approach to LM proves to be effective in improving the probabilities of n-grams while maintaining a large coverage of generic n-grams. In addition, it is common in data mining to transfer information from pre-trained models to new tasks [18]. In literature [19], n-gram LMs are based on the relative frequencies of n-gram events.

Another widely used TL approach to transferring E2E models to a new domain is the fusion of E2E models with an external LM trained on the new domain text data. There are several LM fusion methods, such as shallow fusion, deep fusion [20], and cold fusion [21]. Among them, the simplest and most popular method is shallow fusion, where the external LM is log-linearly interpolated with the E2E model at inference time [22].

However, the shallow fusion approach lacks a definitive probabilistic framework. As an extension, a decoding method has been introduced to facilitate the integration of an external LM with CTC. These methods always used deep learning. A density ratio approach based on Bayes' rule has been proposed for RNN-T in [23]. Another similar model is the Hybrid Autoregressive Transducer (HAT) model [24], which was designed to improve the RNN-T model. Kubo et al. [25] propose a method to improve the performance of E2E speech recognition systems by transferring knowledge from large, pre-trained speech models. In this paper, we have improved the shallow fusion and improved it in the ATC domain.

## 3. Materials and Methods

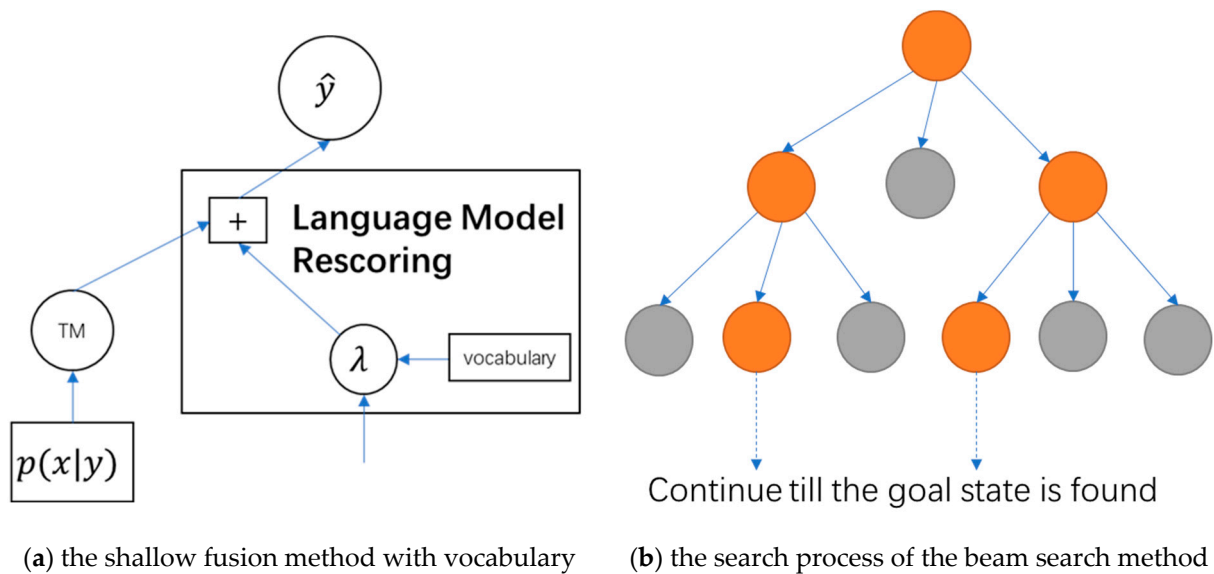
### 3.1. Shallow Fusion

Shallow fusion parallels the conventional use of language models within the decoder of a standard statistical machine translation (SMT) system. At each time interval, the translation model posits a collection of potential words. These candidates are subsequently evaluated based on a weighted sum of the scores provided by both the translation model (TM) and the LM. Using the following equation, the shallow fusion can be distributed.

$$\hat{y} = \underset{y}{\operatorname{argmax}} \log p(y|x) + \lambda \log p_{LM}(y) \quad (1)$$

In Equation (1), the  $x$  and  $y$  are two sequences in sequence-to-sequence (Seq2Seq) models. A basic Seq2Seq model consists of an encoder, which maps an input sequence  $x = (x_1, \dots, x_T)$  into an intermediate representation  $h$ , and a decoder, which in turn generates an output sequence  $y = (y_1, \dots, y_K)$  from  $h$ . The  $\hat{y}$  is the sequence of output for shallow fusion. Usually, the shallow fusion is the last step of ASR, and the sequence  $\hat{y}$  is the text of predicting. Also,  $p(y|x)$  is the probability that the task-specific Seq2Seq model assigns to sequence  $y$  given input sequence  $x$ . The argmax operation is a CTC beam search algorithm in this equation.

Beam search decoding works by iteratively expanding the text hypotheses (beams) with the next possible characters, and only the hypotheses with the highest scores at each time step are retained. An LM can be incorporated into the score computation, and the addition of a lexicon constraint restricts the next possible tokens for the hypotheses so that only words from the lexicon can be generated. The CTC beam search method involves generating the top beam size candidates at each time step while maintaining a set of candidate sequences equivalent to the beam size. The process culminates in the selection of the sequence with the highest probability among the beam-size candidates to produce the final output. Figure 1b shows the beam search process.



**Figure 1.** Graphical illustrations of shallow fusion (a) and beam search (b).

The equation can be separated into two parts. The first part is the equation of the TM. It also shows how the decoder without shallow fusion works:

$$\hat{y} = \operatorname{argmax}_y \log p(y|x) \quad (2)$$

The second part is the equation of LM, which is  $\lambda \log p_{LM}(y)$  in the equation. The  $p_{LM}(y)$  is the language model probability assigned to the label sequence  $y$ . The method of vocabularies in the ASR services mentioned above is using shallow fusion. The vocabularies change the  $\lambda$  to customize the ASR. Figure 1a shows how the vocabulary works in shallow fusion.

The  $\lambda$  is a hyper-parameter that needs to be tuned to maximize the performance of the translation on a development set. We have chosen the n-gram LM as the  $\lambda$ . The n-gram LM is trained on text data, and then it is used in shallow fusion with the CTC beam search decoding to find the best candidates, as shown in Figure 1a. The intuition behind the n-gram model is that instead of computing a character's probability over its entire history, we can approximate history using only recent words.

In the 2-g model for Mandarin, only the conditional probability of the previous character is used to approximate the probability of a character given all the previous characters. The assumption that the probability of a character depends only on the probability of the preceding character is known as the Markov assumption. Markov models are a class of probability models that assume that we can predict the probability of some future entity without looking too far into the past. We can extend the 2-g (which looks one character into the past) to the 3-g (which looks two characters into the past) and thus to the n-gram (which looks n-1 characters into the past). The n-gram algorithm is introduced in Section 3.2. The vocabulary approach is comparative. Our improved method is also based on the vocabulary method.

### 3.2. N-Gram Algorithm

The aim of LM is computing  $P(w|h)$ , the probability of a word  $w$  given some history  $h$ . Suppose the history  $h$  is "CCA 1212" and we want to know the probability that the next word is "connect":  $P(\text{connect}|CCA 1212)$ . One way to estimate this probability is from relative frequency counts: take a very large corpus, count the number of times we see "CCA1212", and count the number of times this is followed by "connect". This would be

answering the question, “Out of the times we saw the history  $h$ , how many times was it followed by the word  $w$ ”, as Equation (3) shows:

$$P(\text{connect}|\text{CCA 1212}) = \frac{C(\text{CCA 1212 connect})}{C(\text{CCA 1212})} \tag{3}$$

With a large enough corpus, we can compute these counts and estimate the probability from Equation (2). While this method of estimating probabilities directly from counts works fine in many cases, it turns out that there is not enough corpus to give us good estimates in most cases.

For this reason, the n-gram is introduced to solve this problem. It separates the history  $h$  into a sequence of  $n - 1$  values as  $X_1, \dots, X_{n-1}$ . If the  $X$  is a word, the  $P(w|h)$  can change to  $P(X_1 = w_1, X_2 = w_2, \dots, X_n = w)$ , simply,  $P(w_1, w_2, \dots, w_{n-1}, w)$ . We used the chain rule of probability:

$$P(X_1 \dots X_n) = P(X_1)P(X_2|X_1)P(X_3|X_{<3}) \dots P(X_n|X_{<n}) = \prod_{k=1}^n P(X_k|X_{<k}) \tag{4}$$

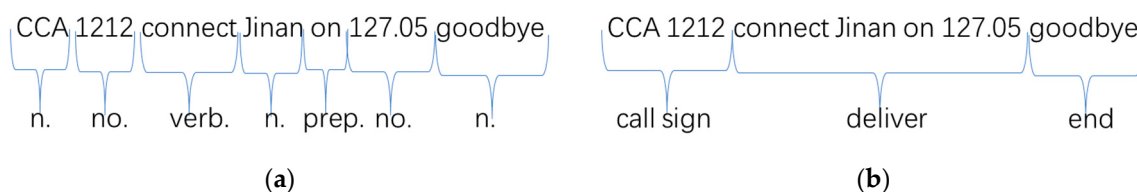
In Equation (4), the  $X_{<k}$  means the sequence of  $X_1, \dots, X_{k-1}$ . The same the  $X_{a:b}$  means the sequence of  $X_a, X_{a+1}, \dots, X_b$ . The n-gram model approximates the probability of a word given all the previous words  $P(w|h)$  by using only the conditional probability of the preceding  $k - 1$  words. The n-gram can be distributed as the following approximation:

$$P(w_k|w_{1:k-1}) \approx P(w_k|w_{k-n+1:k-1}) \tag{5}$$

Equation (5) shows that the bi-gram LM only considers the previous word to predict the next word. In this paper, we expanded the n-gram for PCVCs and used the n-gram LM to replace the LM rescoring part in the shallow fusion.

### 3.3. Improved Shallow Fusion with Extended n-Gram

In Chinese, the n-gram can be used to predict the next character from the previous characters. In the Chinese language, a word is made up of one or more characters. The n-gram can predict the next word from the previous words. In this case, the n-gram is extended from the Chinese characters to the Chinese words. A Chinese structure consists of one or more words. Figure 2 is an example of a sentence and its structure. Figure 2a shows the words in Mandarin. Figure 2b shows the structure as seen by ATC. Similar to extending from characters to words, the n-gram algorithm can also extend to structures.



**Figure 2.** The structure of an ATC instruction. (a) shows the words in English, and (b) shows the structure in the ATC vision.

In our previous work, we tried to generate the instructions for ATC. We found that the Mandarin for ATC is more simple than usual. The structure of the instructions in ATC is almost the same. In our previous work, we used it to generate instructions and in this work, we used it to improve the performance of ASR in ATC.

In this study, we employ a three-layer n-gram model. The initial layer comprises character-based n-grams. For instance, within the tri-gram framework, the probability of the character “A” occurring after the characters “C” and “C” can be represented as  $P(A|CC)$ . The second layer of the n-gram model consists of word-based n-grams. The

word serves as the fundamental unit of Mandarin syntax, as illustrated in Figure 2a. To exemplify, within a tri-gram context, the probability of the word “connect” following the words “CCA” and “1212” can be expressed by  $P(\text{connect}|\text{CCA } 1212)$ . The third layer is dedicated to structure-based n-grams, with exemplary structures depicted in Figure 2b. This can be articulated as  $P(\text{“deliver”}|\text{“call sign”})$  for a bi-gram model. The n-gram for structure predicts the label. Figure 3 shows the n-gram for different parts.

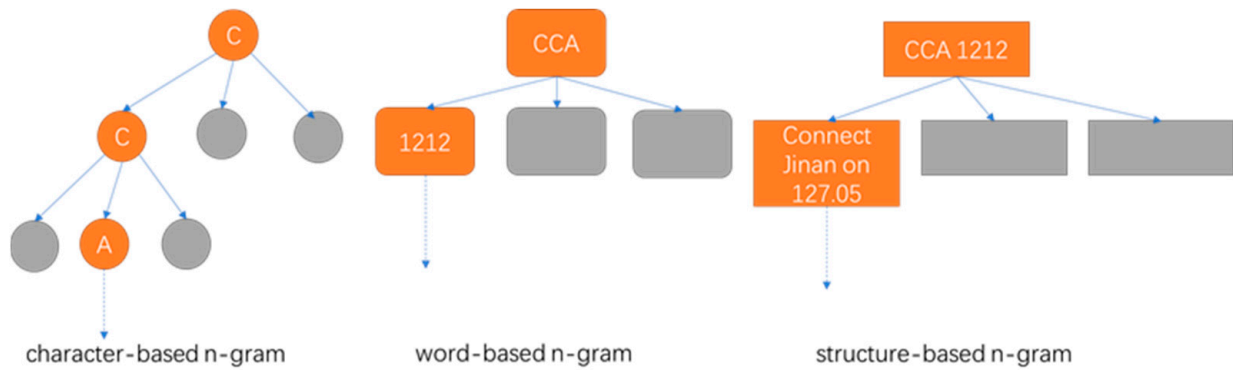


Figure 3. The n-gram for different parts of the utterance.

The structure is made up of words, and the words are made up of characters. Figure 4 shows the architecture of the 3-layer n-gram and how a higher layer is built step by step. The red lines of feedback show how the upper layer influences the lower. The feedback is a kind of superficial fusion, where the upper layer is seen as a scoring part and the lower layer is seen as a TM.

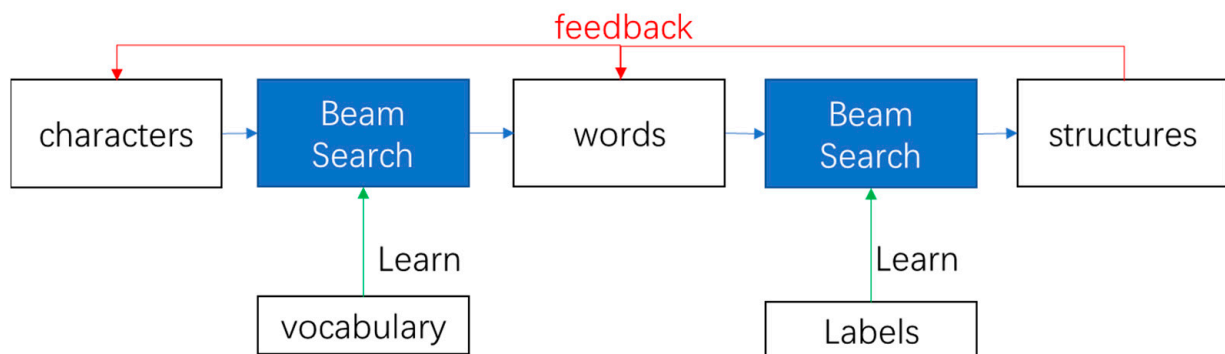


Figure 4. The architecture of a 3-layer n-gram.

The feedback between the words and the structures is a little bit different. The structure layer is working on the labels, and the feedback from it is to the labels as well. The labels are the classifications of words and their structures. The prediction of n-gram structure is for labels. The history of the n-gram in the structure is also a sequence of the labels. The n-gram in the structure uses the sequences of the labels to predict the next label. One or more labels of words make up the label of structure. The feedback of the structure is the limitation of the words.

The example above is in English. Mandarin has more challenges. For example, there is no space between two words, and the number of characters in different structures is almost the same. This n-gram model is only used for PCVCs. Firstly, PCVCs are based on half-duplex FM, which means that only one person can speak at a time. This also creates PCVCs without long and continuous conversations. Secondly, the structure of PCVCs is simpler than usual, and there are fewer objects in ATC. Finally, it is impossible to find a new structure in ATC instructions. An LM can contain any structure and all models for PCVCs.

### 3.4. ASR of Testing

It is common to use perplexity to test the LM. However, this paper aims to use the LM to improve the performance of ASR. An ASR system for testing is still needed for the study. We intended to choose the public ASR systems that are training-free. However, the custom LM cannot be accepted by these training-free ASR models from large companies. They only accept data for ASR adaptation. For these reasons, the training of an ASR model was the focus of the study. Section 4 describes the details of the training process. This section presents the structure of the ASR model.

In this paper, we used the DeepSpeech2-based ASR system to test our method [26]. DeepSpeech2 is an open source E2E ASR system [27]. Figure 5 shows the architecture of the DeepSpeech2 ASR system.

At the bottom of Figure 4 is Feature Extraction. In Feature Extraction, each utterance  $x$  is a time series of length  $T$  where each time slice is a vector of audio features,  $x^t, t = 1, 2, \dots, T$ .

The next part is the encoder, which is made up of two subsampling convolutional layers and a multi-RNN layer. The goal of the subsampling layer is to extract local features, reduce the number of frames input to the model, reduce the amount of computation, and facilitate model convergence. The goal of the RNN is to convert an input sequence  $x$  into a final transcription  $y$ .

The last part is the decoder. The primary role of the decoder is to decode the probability output from the encoder into the final text result. The softmax layer maps the feature vector to a vector of length equal to the size of the vocabulary and stores the probability that the result of the current step should be predicted for each word in the vocabulary. For the CTC decoder, there are three main algorithms: greedy search, beam search, and prefix beam search. The main work of CTC is to align the time slice  $x^t$  and the label  $y$ .

As shown in Figure 1a, the shallow fusion input is  $p(y|x)$ , which is the same as the output of the softmax layer in DeepSpeech2. We replaced the CTC layer in DeepSpeech2 with the shallow fusion method. We used the LSTM for the RNN cell and built 2 subsampling layers and five layers of the LSTM ASR system, represented as “2Cov+5LSTM”.

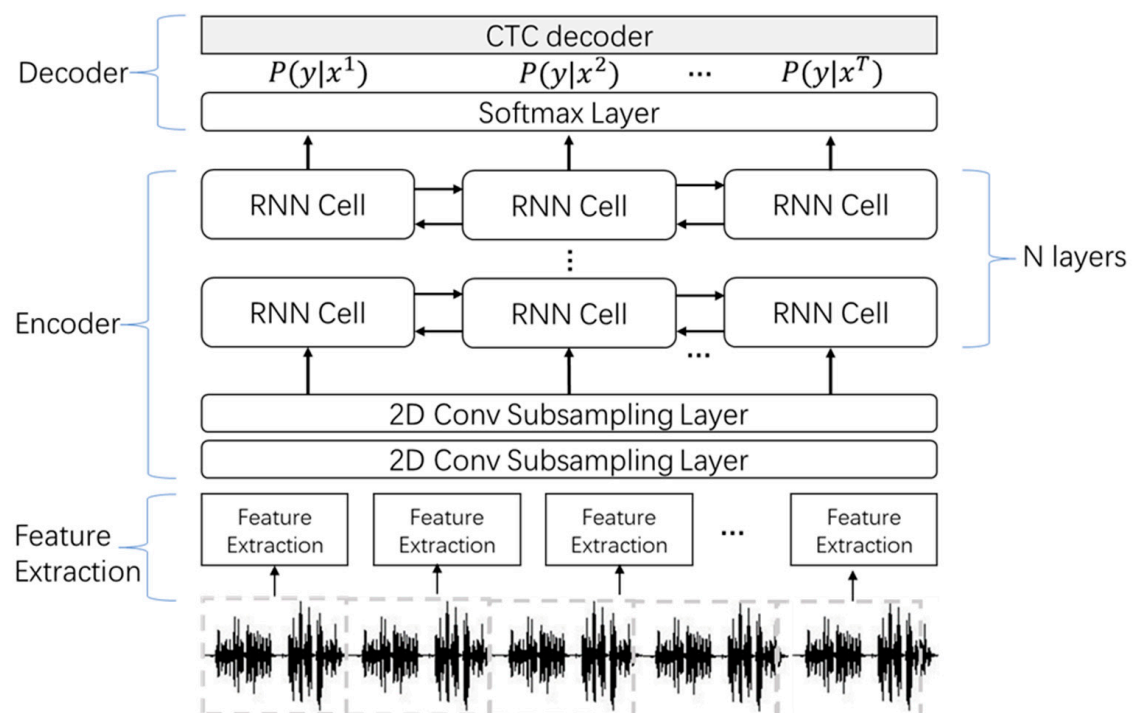


Figure 5. The architecture of DeepSpeech2.

## 4. Experiment and Results

### 4.1. Setup

The fusion method was tested and improved on the speech recognition task using PCVC data in this study. The final performance on the speech recognition task was evaluated in CER in these experiments. The CER was calculated according to the following rule:

$$CER = \frac{I + D + S}{N} \times 100\% \quad (6)$$

where the denominator  $N$  represents the total length of the true label and the notation  $I$ ,  $D$ ,  $S$  denote the number of the insertion, deletion, and substitution operations, respectively. In addition, the perplexity was applied to evaluate the performance of LMs. For a test set  $W = w_1w_2 \dots w_N$ , the perplexity can be represented as:

$$\text{perplexity}(W) = P(w_1w_2 \dots w_N)^{-\frac{1}{N}} \quad (7)$$

where  $P(w_1w_2 \dots w_N)$  is the probability of the sequence  $w_1w_2 \dots w_N$ . In this study, the  $W$  is a set of characters. The character-based perplexity includes the word-based perplexity because the character is the basic part of Chinese. In addition, the LMs in this study used the character-based n-gram in the bath and the word-based n-gram. The use of character-based perplexity can work for all LMs of Chinese.

We chose Aishell as our training data and control group. Aishell is a 178-h-long Mandarin speech corpus. The manual transcription accuracy rate is over 95%, thanks to professional speech annotation and strict quality control. The corpus is divided into training, development, and test sets. The database of PCVCs is derived from the first-line control recordings of the North China air traffic control Bureau of Civil Aviation of China and the control simulation training recordings of the Civil Aviation Flight University of China. This database contains 35,358 utterances.

To train the LMs of shallow fusion and enhanced fusion with extended n-gram, we need some text data. We use the text data from the Aishell corpus. We also need some text data for ATC instructions. The text database of ATC instructions is provided by our previous work, which generates the ATC instructions. The text of Aishell and the text data of PCVCs are very different. Table 1 shows the results of training character-based recurrent neural network language models on each of the datasets and evaluating both datasets. Note that the model trained on Aishell performs poorly on PCVCs and vice versa, indicating that the two corpora are very different. For this reason, ASR models trained on Aishell will perform poorly on PCVCs.

**Table 1.** The perplexities for character RNN LMs trained and tested on a different dataset.

Training Database	Perplexity in Aishell	Perplexity in PCVCs
Aishell	2.742	4.630
PCVCs	5.481	3.735

Table 1 is intended to show how the characters differ between the two corpora. The difference in perplexity can stand for the difference in both the proportion of the characters and the sequence of the characters in two corpora.

### 4.2. Training

In this study, ASR and LM were trained separately. For the ASR, the input sequence consisted of 40 mel-scaled filterbank features. We extended the data sets with a noise augmentation; random background noise is added with a probability of 40% and a uniform random SNR between 0 and 15 dB. No other form of regularization was used.

For LM, the input was some text data from the Aishell corpus. In the study, we used the same ASR system and the same LM. To compare with the traditional method, after



the training, we added fusion methods to the LM and obtained three different LMs. The first LM is trained by the Aishell corpus without any fusion. This is a control group. The second LM is based on the shallow fusion of vocabularies. The fusion of vocabulary is also the main method provided by cloud service providers to customize their ASR systems. The third LM is fused by ATC instructions, which use the fusion method as we described before. To compare with other n-gram methods for the fusion method, word-based and character-based n-grams are included in the test. This is to show that our method is better than the usual n-gram in the ATC domain. In this experiment, all n in n-gram is 2, meaning we use bi-gram for the experiment.

Some text data from the Aishell corpus was used as input for the LM. In the investigation, the same ASR system and the same LM were used. For comparison with the traditional method, after the training, we added fusion methods to the LM and obtained three different LMs. The first LM was trained by the Aishell corpus without any fusion. The first LM was trained without fusion as a control group. The second LM was based on the shallow fusion of vocabularies. The fusion of vocabulary is also the main method that cloud service providers use for the customization of their ASR systems. The third LM, which used the fusion method described above, was fused according to ATC instructions. To show that our method is better than the usual n-gram in the ATC domain, we added word-based and character-based n-grams to the test for comparison with other n-gram fusion methods.

The perplexity of different LMs on different corpora is shown in Table 2. It can be seen that the fusion methods have improved the perplexity of the LMs on the Aishell corpus. The fusion methods increase perplexity in the source domain (Aishell corpus) and decrease it in the target domain (PCVCs corpus). The lower the perplexity of a model on the data, the better the model, and the minimization of the perplexity is equivalent to the maximization of the test set probability according to the language model. In addition, the 3-layer bi-gram still has better performance compared to other bi-gram methods.

**Table 2.** The perplexities for LMs with different fusion methods.

Fusion Method	Perplexity in Aishell	Perplexity in PCVCs
Null	2.619	5.352
vocabulary	3.174	3.428
3-layer bi-gram	3.813	2.185
Word-based bi-gram	3.716	2.793
Character-based bi-gram	3.324	3.478

Fusion is a transfer learning method used to transfer domains. In theory, fusion aims to improve models in the targeted domain without changing performance in the source domain. In practice, however, the perplexity is calculated according to the frequency. The weight of the source domain is reduced by adding the vocabularies and ATC instructions. This means that merging makes the perplexity in the Aishell corpus larger. For the vocabulary, only the words in the database are used for the fusion, ignoring other information from the ATC instructions. The n-gram methods can obtain more information from the ATC database. Therefore, they perform better than the vocabulary method. The only exception is that the vocabulary method performed better than the character-based bi-gram. This is because vocabulary also works on characters, but the character-based bi-gram also counts nearby characters even if there is no relationship between them.

#### 4.3. Results

The CER of the “2Cov+5LSTM” ASR system with different LMs is shown in Table 3. Fusion had an increase in CER in the source domain and a decrease in CER in the target domain. It can also be seen from Table 2 that the fusion methods had a similar CER in the two domains.

**Table 3.** The CER for ASR and LM with different fusion methods.

Fusion Method	CER in Aishell	CER in PCVCs
Null	0.1453	0.3493
vocabulary	0.1526	0.2876
3-layer bi-gram	0.1579	0.2761
Word-based bi-gram	0.1524	0.2918
Character-based bi-gram	0.1613	0.2837

Slightly better than the vocabulary fusion was the 3-layer bi-gram fusion. The amount of data in the two fusions is, however, significantly different. The number of words in the instruction data is much larger than the number of words in the vocabulary data. There were 3000 instructions used for the fusion. The same data from the 3000 instructions was also used for the vocabulary fusion. Without numbers and times, the vocabulary was made up of the words from the 3000 instructions. There were only 376 words in the vocabulary for fusion after cleaning the same words. However, compared to vocabulary fusion, our improved fusion did not perform as well on CER as it did on perplexity. The perplexity of the fusion shows that the improved fusion adopted a better LM. This method performed as well in ASR as the traditional method.

The 3-layer bi-gram still performed better than the other fusion methods for the word-based bi-gram and the character-based bi-gram. The character-based bi-gram performed better than the word-based bi-gram because of our explanation. According to our explanation, the word-based bi-gram contains more information than the character-based one. Therefore, it would perform better.

## 5. Discussion

DeepSpeech2 is an ASR system that is open source. Some trained models for DeepSpeech2 use the same training corpus as we used to train ours. But the CER for their ‘2Cov+5LSTM’ model was 0.0666. It did much better than ours. Replacing the CTC decoder with LM is the biggest difference. There are many ways to improve ASR before training, during training, and after training: data augmentation and fine-tuning. These pre-training and in-training methods can be ignored by using transfer learning and adaptation. It can save a lot of work. This is the first reason why it is so important to use customization.

In recent years, the large model has become very popular. The large model has also been introduced into the ASR service by the cloud service providers. The ASR service becomes more universal and robust with more large training data. The second reason is that the robust ASR can solve the problem of noise and the high speech rate in air traffic control. A universal ASR system can also recognize the terminologies in ATC. Adapting the ASR can help emphasize the ATC domain. The challenge of noise and high speech rates is not only in the ATC domain. However, the terminology of ATC may be used only in ATC. This means that it is more important to adapt the LM in ASR.

Previous researchers have tried to solve the challenges of ATC. These include complex noise, excessive speech rate, code-switching, terminology, and accented speech. Our focus is on the advantages of ATC speech. ATC speech is a working language. It emphasizes the accuracy of semantics, the conciseness of grammar, and the clarity of pronunciation. Instead of “CCA1212Connect Jinan on 127.05, goodbye”, controllers might say “Connect Jinan on 127.05, goodbye, CCA1212”. This only changes the order of the structures. It does not change the order within the structure. Our improved shallow fusion works well if the language has some words or their labels in a fixed order. This order has always appeared in the database.

Other layers of n-grams, such as utterance-based n-grams and flight-process-based n-grams, could be built for future research. Flight processes are ordered. For example, the process of landing comes after the process of take-off, because flight processes are physical. In addition, the utterances are based on the processes of the flight. The process of flight determines the utterances. Building layers of utterances and processes can be a big

improvement. But there are some problems with it. First, to build the utterance layer: Each utterance needs a label. Labeling utterances is based on labeling structures, and labeling structures are based on labeling words. Each upper label is composed of one or more lower labels. There may be some problems with more layers of labels. Secondly, controllers do not talk to just one flight. There are many cases where one utterance is intended for one flight and the next utterance is intended for a different flight. But the sequence of utterances and processes is different for each flight. For each flight, we have to separate the utterances.

## 6. Conclusions

An improved fusion method for LM was developed in this study. This method is a transfer learning method of fusion. It made the LM work in the ATC domain. Compared to the traditional fusion using vocabulary, the improved fusion uses ATC instructions (sentences), which can contain more information. The n-gram of the traditional fusion can only use some characters to form a word. For improved fusion, we have divided the n-gram into three layers. The top is the sentence, which is composed of a structure of instructions. The middle is the structure composed of some words. The bottom is the word composed of characters. The perplexity of the LMs shows that the LM of the improved fusion is the best LM of the three. We then add an ASR system to test the performance of the LMs. The CER without fusion is 0.3493 in the ATC corpus. After traditional fusion, the CER drops to 0.2876. In addition, the CER of our method is 0.2761. The difference in CER between traditional and improved fusion is less than 2%.

Compared to other n-gram methods, the 3-layer n-gram performed better. This means that our improved fusion with extended n-gram was a better LM and performed better in the ATC domain. We believe that our method will perform much better after adding more layers in the ATC domain in our future research. It shows the relationship between utterances and processes after adding more layers. The relationship between utterances and processes may work for language understanding in the ATC domain if we extend the n-gram method to utterances and processes.

**Author Contributions:** Conceptualization, J.F. and W.P.; methodology, J.F.; software, J.F.; validation, J.F. and W.P.; formal analysis, J.F. and W.P.; investigation, W.P.; resources, J.F.; data curation, J.F.; writing—original draft preparation, J.F.; writing—review and editing, W.P.; visualization, J.F.; supervision, W.P.; project administration, W.P.; funding acquisition, W.P. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by the National Key R&D Program of China grant number NO. 2021YFF0603904.

**Data Availability Statement:** Data available on request due to restrictions, e.g., privacy or ethics.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## Abbreviations

The following abbreviations are used in this manuscript:

ATC	Air Traffic Control
ATCO	Air Traffic Controller
PCVC	Pilot-Controller Voice Communication
ASR	Automatic Speech Recognition
AM	Acoustic Model
PM	Pronunciation Model
LM	Language Model
TM	Translation Model
E2E	End-to-End
TL	Transfer Learning
RNN	Recurrent Neural Network
DNN	Deep Neural Network

HMM	Hidden Markov Model
MFCC	Mel Frequency Cepstral Coefficient
CTC	Connectionist Temporal Classification
RNN-T	Recurrent Neural Network Transducer
SMT	Statistical Machine Translation
CER	Character Error Rate

## References

- Hawkins, F.H. *Human Factors in Flight*, 2nd ed.; Routledge: London, UK, 1993; pp. 1–11.
- de Cordoba, R.; Ferreiros, J.; San-Segundo, R.; Macias-Guarasa, J.; Montero, J.M.; Fernandez, F.; D'Haro, L.F.; Pardo, J.M. Air traffic control speech recognition system cross-task and speaker adaptation. *IEEE Aerosp. Electron. Syst. Mag.* **2006**, *21*, 12–17. [[CrossRef](#)]
- 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](#)]
- 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](#)]
- Kheddar, H.; Himeur, Y.; Al-Maadeed, S.; Amira, A.; Bensaali, F. Deep transfer learning for automatic speech recognition: Towards better generalization. *Knowl. Based Syst.* **2023**, *277*, 110851. [[CrossRef](#)]
- Lin, Y.; Yang, B.; Li, L.; Guo, D.; Zhang, J.; Chen, H.; Zhang, Y. ATCSpeechNet: A multilingual end-to-end speech recognition framework for air traffic control systems. *Appl. Soft Comput.* **2021**, *112*, 107847. [[CrossRef](#)]
- Lin, Y.; Tan, X.; Yang, B.; Yang, K.; Zhang, J.; Yu, J. Real-time Controlling Dynamics Sensing in Air Traffic System. *Sensors* **2019**, *19*, 679. [[CrossRef](#)] [[PubMed](#)]
- Watanabe, S.; Hori, T.; Kim, S.; Hershey, J.R.; Hayashi, T. Hybrid CTC/attention architecture for end-to-end speech recognition. *IEEE J. Sel. Top. Signal Process.* **2017**, *11*, 1240–1253. [[CrossRef](#)]
- Sainath, T.N.; He, Y.; Li, B.; Narayanan, A.; Pang, R.; Bruguier, A.; Chang, S.-Y.; Li, W.; Alvarez, R.; Chen, Z.; et al. A streaming on-device end-to-end model surpassing server-side conventional model quality and latency. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, Barcelona, Spain, 4–8 May 2020; pp. 6059–6063.
- Li, J.; Zhao, R.; Meng, Z.; Liu, Y.; Wei, W.; Parthasarathy, S.; Mazalov, V.; Wang, Z.; He, L.; Zhao, S.; et al. Developing RNNT models surpassing high-performance hybrid models with customization capability. In Proceedings of the Interspeech, Shanghai, China, 25–29 October 2020; pp. 3590–3594.
- Azure AI Speech. Available online: <https://azure.microsoft.com/products/ai-services/ai-speech/> (accessed on 1 February 2024).
- Speech To Text. Available online: <https://aws.amazon.com/transcribe/> (accessed on 1 February 2024).
- ASR Customization. Available online: <https://www.huaweicloud.com/intl/en-us/product/asrc.html> (accessed on 1 February 2024).
- Basics of Speech Recognition and Customization of Riva ASR. Available online: <https://docs.nvidia.com/deeplearning/riva/user-guide/docs/asr/asr-customizing.html> (accessed on 1 February 2024).
- Pan, S.J.; Yang, Q. A Survey on Transfer Learning. *IEEE Trans. Knowl. Data Eng.* **2010**, *22*, 1345–1359. [[CrossRef](#)]
- Mridha, M.F.; Ohi, A.Q.; Hamid, M.A.; Monowar, M.M. A study on the challenges and opportunities of speech recognition for Bengali language. *Artif. Intell. Rev.* **2022**, *55*, 3431–3455. [[CrossRef](#)]
- Jiang, D.; Tan, C.; Peng, J.; Chen, C.; Wu, X.; Zhao, W.; Song, Y.; Tong, Y.; Liu, C.; Xu, Q.; et al. A GDPR-compliant ecosystem for speech recognition with transfer, federated, and evolutionary learning. *ACM Trans. Intell. Syst. Technol.* **2021**, *12*, 1–19. [[CrossRef](#)]
- Devlin, J.; Chang, M.-W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv* **2018**, arXiv:1810.04805. [[CrossRef](#)]
- Deena, S.; Hasan, M.; Doulaty, M.; Saz, O.; Hain, T. Recurrent neural network language model adaptation for multi-genre broadcast speech recognition and alignment. *IEEE/ACM Trans. Audio Speech Lang.* **2018**, *27*, 572–582. [[CrossRef](#)]
- Gulcehre, C.; Firat, O.; Xu, K.; Cho, K.; Barrault, L.; Lin, H.-C.; Bougares, F.; Schwenk, H.; Bengio, Y. On using monolingual corpora in neural machine translation. *arXiv* **2015**, arXiv:1503.03535. [[CrossRef](#)]
- Sriram, A.; Jun, H.; Sathesh, S.; Coates, A. Cold fusion: Training seq2seq models together with language models. In Proceedings of the Interspeech, Hyderabad, India, 2–6 September 2018; pp. 387–391.
- Toshniwal, S.; Kannan, A.; Chiu, C.-C.; Wu, Y.; Sainath, T.N.; Livescu, K. A comparison of techniques for language model integration in encoder-decoder speech recognition. In Proceedings of the IEEE Spoken Language Technology, Athens, Greece, 18–21 December 2018; pp. 369–371.
- McDermott, E.; Sak, H.; Variani, E. A density ratio approach to language model fusion in end-to-end automatic speech recognition. In Proceedings of the IEEE Automatic Speech Recognition and Understanding, Singapore, 14–18 December 2019; pp. 434–441.
- Variani, E.; Rybach, D.; Allauzen, C.; Riley, M. Hybrid autoregressive transducer (HAT). In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, Barcelona, Spain, 4–8 May 2020; pp. 434–441.

25. Kubo, Y.; Karita, S.; Bacchiani, M. Knowledge transfer from large-scale pretrained language models to end-to-end speech recognizers. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, Singapore, 23–27 May 2022; pp. 8512–8516.
26. Amodei, D.; Ananthanarayanan, S.; Anubhai, R.; Bai, J.; Battenberg, E.; Case, C.; Casper, J.; Catanzaro, B.; Cheng, Q.; Chen, G.; et al. Deep Speech 2: End-to-end speech recognition in English and Mandarin. In Proceedings of the 33rd International Conference on International Conference on Machine Learning, New York, NY, USA, 19–24 June 2016; pp. 173–182.
27. GitHub-Mozilla/DeepSpeech. Available online: <https://github.com/mozilla/DeepSpeech> (accessed on 1 February 2024).

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.