aerospace

Article

A Context-Aware Language Model to Improve the Speech Recognition in Air Traffic Control

Dongyue Guo [1], Zichen Zhang [2], Peng Fan [1], Jianwei Zhang [2,*] and Bo Yang [2,*]

[1] National Key Laboratory of Fundamental Science on Synthetic Vision, Sichuan University, Chengdu 610000, China; dongyueguo@stu.scu.edu.cn (D.G.); fanpeng2019@stu.scu.edu.cn (P.F.)
[2] College of Computer Science, Sichuan University, Chengdu 610000, China; zhang_zichen@stu.scu.edu.cn
* Correspondence: zhangjianwei@scu.edu.cn (J.Z.); boyang@scu.edu.cn (B.Y.)

Abstract: Recognizing isolated digits of the flight callsign is an important and challenging task for automatic speech recognition (ASR) in air traffic control (ATC). Fortunately, the flight callsign is a kind of prior ATC knowledge and is available from dynamic contextual information. In this work, we attempt to utilize this prior knowledge to improve the performance of the callsign identification by integrating it into the language model (LM). The proposed approach is named context-aware language model (CALM), which can be applied for both the ASR decoding and rescoring phase. The proposed model is implemented with an encoder–decoder architecture, in which an extra context encoder is proposed to consider the contextual information. A shared embedding layer is designed to capture the correlations between the ASR text and contextual information. The context attention is introduced to learn discriminative representations to support the decoder module. Finally, the proposed approach is validated with an end-to-end ASR model on a multilingual real-world corpus (ATCSpeech). Experimental results demonstrate that the proposed CALM outperforms other baselines for both the ASR and callsign identification task, and can be practically migrated to a real-time environment.

Keywords: language model; automatic speech recognition; air traffic control; dynamic contextual information

check for updates

Citation: 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. https://doi.org/10.3390/aerospace8110348

Academic Editors: Alexei Sharpanskykh and Yi Lin

Received: 1 October 2021
Accepted: 8 November 2021
Published: 16 November 2021

1. Introduction

In the past few decades, the automatic speech recognition (ASR) technique has made great processes by data-driven methods. It has been widely used in various fields as one of the important interfaces for human–machine interaction, such as air traffic control (ATC), mobile devices. Currently, in the ATC procedure, the speech communication and ATC system support the ATC operation together to ensure its efficiency and safety. On the one hand, the air traffic controller (ATCO) issues speech instructions via the very high frequency (VHF) radio, whereas the pilot subsequently read the instructions back. On the other hand, flight plans, aircraft positions provided by surveillance radar, and other contextual information are integrated into the terminal of the ATC system to assist ATCO in managing the airspace.

However, due to technical limitations, ATC speech communication is independent of the ATC system, which fails to understand the real-time traffic dynamics. Thus, the ASR system becomes a promising technique to bridge the speech communication and ATC system. Recently, more and more attention has been paid to employ the ASR techniques to empower ATC applications, such as the ATC assistance system [1], operational safety monitoring system [2,3], and the ATCO training system [4,5].

In the above-mentioned applications, the flight callsign is the only correlation between ATC speech and real-time contextual information of the ATC system. In general, only the ASR results with a correct callsign can be applied to the downstream applications [6].

Therefore, improving the performance of the callsign identification is the key to advance the ASR technique into industrial application.

Exploring the ASR techniques in the field of ATC communications has attracted increasing interest in recent years. The techniques and challenges in the ATC-related research were reviewed in [7,8]. A cascaded framework was studied to cope with the multilingual and out-of-vocabulary (OOV) issues in the ATC domain [9]. An exploratory benchmark of several advanced ASR models trained on ATC corpus was presented in [10]. Semi-supervised Learning [11] and representation learning [12,13] approaches were also introduced to leverage abundant untranscribed speech data to improve ASR performance in the ATC domain. Furthermore, an ASR and callsign detection challenge of the ATC was held by the Airbus company in 2018 [14].

Although significant progress of the ASR performance has been made in the ATC domain [9–15], recognizing isolated digits of the callsign is a challenging task in the ATC domain due to their widespread usage and ambiguous meanings [8]. For example, an ATC instruction *Air China four four one climb maintain eight thousand one hundred meters* contains multiple digits, the *four four one* is a part of the callsign while the *eight thousand one* refers to the flight level. The best result of the callsign detection F1-score reported in [14] is about 83% in the AIRBUS-ATC [16] corpus, whereas it is about 74% accuracy for another multilingual ASR system [15]. Fortunately, the flight callsign is a kind of prior ATC knowledge and available from the contextual information, such as surveillance radar and flight plan. In other words, if the callsign entity in the dynamic contextual information can be encoded into a text set, the callsigns involved in the ATC speech are most possibly one of the elements. Intuitively, integrating the contextual information into the ASR system is expected to be an effective way to improve the performance of the callsign identification.

In this work, we attempt to utilize this prior knowledge (flight callsign) to improve the performance of the callsign identification in the ASR system. To this end, a context-aware language model (CALM) is proposed to integrate the contextual information into the language model (LM). Moreover, as shown in Figure 1, a contextual ASR system is designed to integrate the CALM and end-to-end acoustic model (AM). Compared with conventional LM, the core idea of the proposed approach is to bias the output of the AM using CALM which can consider the embedding of the dynamic contextual information. Furthermore, the CALM is incorporated into the ASR system in two ways, i.e., decoding with beam search and rescoring based on the N-best list.

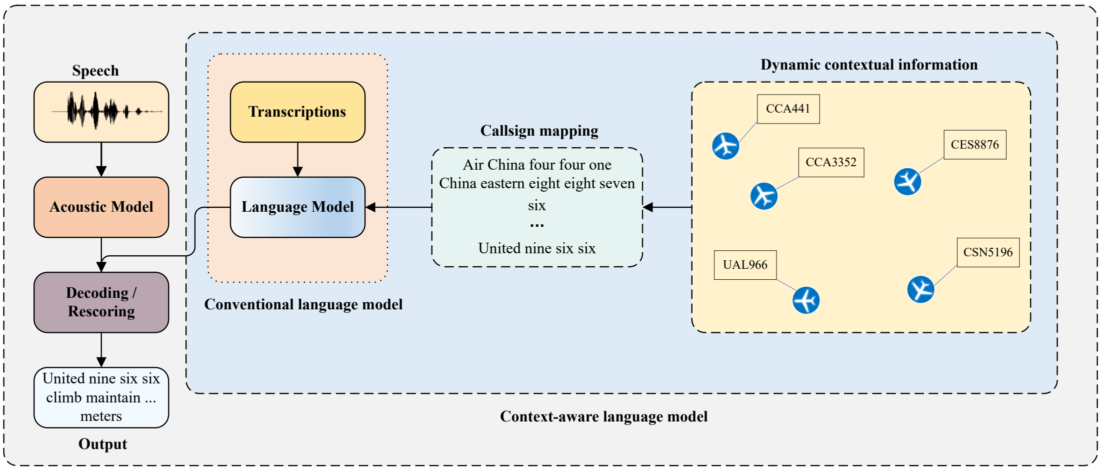


Figure 1. The architecture of contextual ASR system using CALM.

In general, the proposed CALM is implemented with an encoder–decoder architecture, in which the encoder module consists of the text encoder and context encoder. To consider the prior callsign set in the contextual information, an extra encoder module, i.e., context encoder, is proposed to convert the callsign into text-related representations. A shared embedding layer is designed to learn common correlations of the input tokens between the text encoder and context encoder. To discriminate the contributions of the AM output and the predefined callsigns, the context attention mechanism is also designed to support the decoder module to generate the final ASR result. In addition, a callsign mapping strategy is innovatively proposed to consider the multilingual in the ATC speech and the multi-callsign entities in the context. Finally, a simple yet effective context simulation method is developed to complete the modeling training on the existing corpus, which further supports the real-time applications.

By combining with an end-to-end ASR model, the proposed approach is validated on a real-world multilingual speech corpus, i.e., ATCSpeech [15]. Experimental results demonstrate that the proposed CALM outperforms other baselines, which not only shows desired performance improvement on the ASR task (about 4.36% character error rate), but also achieves about 20% accuracy improvement for the callsign identification. Most importantly, the efficiency and effectiveness of the proposed approach are also confirmed on a 5-h real environment dataset, in which both the ATC speech and the contextual information were collected from Chengdu area control.

In summary, the main contributions of this work are as follows:

- A novel neural network language model, called CALM, is proposed to improve the callsign identification in ATC-related ASR systems.
- Compared to conventional LM, the proposed CALM has the ability to integrate the contextual information into the LM decoding by the designed context encoder and context-aware decoder, which improves the ASR performance from the perspective of scene awareness.
- To fuse the representations of the text and the contextual information, a context attention mechanism is proposed to generate a joint representation vector that further supports the context-aware decoder.
- We integrate the CALM into the decoding and rescoring procedure of the ASR systems and validate on the real-world speech corpus.

The remainder of the paper is organized as follows: the previous works of the contextual ASR are briefly reviewed in Section 2. Section 3 presents the architecture of the AM and CALM for constructing the ATC ASR system in this work. In Section 4, we evaluate the proposed CALM in terms of character error rate and callsign accuracy on both decoding and rescoring procedures. The conclusion and future work are described in Section 5.

2. Related Work

2.1. Contextual ASR Systems

Integrating contextual information into the ASR system to improve performance has been studied in both conventional hybrid and end-to-end systems. In general, there are three ways of integrating contextual knowledge into ASR systems, i.e., weighted finite-state transducer (WFST) based decoding, developing external LM, and end-to-end contextual ASR model.

In the HMM-based ASR system, the context information is usually injected into the main finite-state transducer (FST) graph to support the decoding by a WFST [17]. In [18], the lexicon and grammar served as straightforward extensions to generate the recognition search space by on-the-fly composition and delay construction mechanism. A biasing WFST method composed a baseline WFST and a compact WFST representation of the contextual n-grams was used for a voice search application [19].

An on-the-fly rescoring mechanism was proposed to adjust the LM weights of n-grams which is relevant to the dynamic context during the decoding procedure in [20]. In [21], the class LM and word mapping algorithm were proposed to achieve the rare entity words recognition with the LAS (Listen, Attend, and Spell) [22] architecture. A shallow-fusion end-to-end biasing method [23] showed the competitive performance with the recurrent neural network transducer (RNN-T) [24] model.

End-to-end contextual ASR models incorporate contextual information into the recognition process by a single neural network. A contextual-LAS (CLAS) architecture was proposed to consider contextual information by an all-neural mechanism and outperform online rescoring techniques [25]. To improve the recognition for entity names, an end-to-end contextual RNN-T model was presented in [26] for open domain ASR.

Overall, these methods are able to improve the performance of recognizing proper nouns and personalized user vocabulary of the contextual information to a certain extent. It can be found that the contextual ASR systems tends to be developed from the external components to end-to-end manner. However, the end-to-end model often requires a large of samples in the training process. Developing an external LM for integrating contextual information is still a popular technique in many applications.

2.2. ATC Related Works

Due to a wealth of contextual information in the ATC environment, various studies attempted to utilize contextual information to improve the performance of the ATC-related ASR system in recent years [27,28]. A knowledge-based lattice rescoring method [29] was investigated to rescore the ASR hypothesis by a dynamic weighted constraint satisfaction function with dynamic contextual information. The knowledge of the dynamic contextual information was extracted by the ATC grammars which were specified by the International Civil Aviation Organization (ICAO). In [30], the contextual information is generated from a planning system, in which a grammar WFST based approach was further proposed to improve the ASR performance. The ASR hypothesis was also updated by a weighted Levenshtein distance of all possible words that are produced by an additional sequence labeling system [31].

As it can be seen, the WSFT is a standard component of the above methods which highly rely on an external module to generate the required contextual information. Inspired by the success of Deep Fusion [32] and Cold Fusion [33] methods, we attempt to develop a context-aware LM using the deep fusion-based method and integrate it into the ASR system. Specifically, instead of processing the contextual information separately, the proposed approach understands them in a fused and straightforward manner by a neural architecture.

3. Methodology

3.1. The Acoustic Model

Considering that the end-to-end ASR systems are often the most efficient method and deliver competitive quality in recent years [12,22,24,34], a connectionist temporal classification (CTC) based model referring to Deepspeech 2 [35] is introduced to serve as the AM in this work. In general, the AM model consists of convolutional neural networks (CNN), recurrent neural network (RNN), and fully connected (FC) layers. The spectrogram of the speech extracted by a series of linearly spaced log-filterbanks filters served as the model input. Then, three Conv1D layers are stacked to aggregate the local frequency dependencies between the adjacent speech frame and learn high-level representations. Seven bi-directional RNN layers with gated recurrent units (GRU) are applied to capture the long-term temporal dependencies. In addition, the FC layer outputs the probability of given tokens condition on the input speech frame-wisely. Finally, the training error is evaluated by the CTC criterion [36] to further upgrade the training parameters.

In this work, the spectrogram dimension of the input is set to 81 with 25 ms windows and 15 ms overlaps. The CNN channels, filter size, and stride are set to (512, 512, 512), (5, 5, 5), and (1, 1, 2), respectively. These parameters benefit the reduction in the size of output while retaining sufficient receptive fields of the CNN. Furthermore, the BatchNorm1D layer and Hardtanh activation are employed to transform the output features of each Conv1D layer. All RNN layers adopt 512 neurons, which is consistent with the dimension of features output by Conv1D layers. In the training process, the Adam optimizer with an initial learning rate of 10^{-4} is applied to train the AM. An early stopping strategy is performed to terminate the training procedure by observing the validation loss.

3.2. Context-Aware Language Model

With the speech signal being X and a word sequence being W, the target of the ASR task can be described as:

$$\begin{aligned} W^* &= \arg\max_W P(W \mid X) \\ &= \arg\max_W \frac{P(X \mid W)P(W)}{P(X)} \\ &= \arg\max_W P(X \mid W)P(W) \end{aligned} \tag{1}$$

for which $P(X \mid W)$ is predicted by the acoustic model, while the language model aims to build the correlation distribution of the word sequences W.

LM is a powerful way to improve the ASR performance by building vocabulary correlations from numerous existing corpora. However, in practice, the probability of a word sequence is determined by both the historical experience and real-time contextual information. The former generally consists of phrases, fixed terms, grammar, and other customed rules, while the latter mainly focuses on the information that may be affected by real-time contexts, such as the personalized data on the mobile devices and the flight callsign in the ATC environment.

Intuitively, for a certain application, incorporating contextual information into the ASR system is a promising way to improve its final performance. To this end, a novel perspective is introduced to utilize the contextual information empowered LM. Thereby, the target of the LM is refined as $P(W \mid C)$, where C is the real-time context vector.

The proposed model is called CALM, whose architecture is illustrated in Figure 2. In general, the model consists of three modules, including text encoder, context encoder, and context-aware decoder. The detailed descriptions of the three modules are described as follows:

- **Text Encoder:** The text encoder is composed of an input layer, embedding layer, and several LSTM layers. The purpose of the text encoder is to convert the input sequence into high-level feature representations. For a text sequence $W = \{w_1, w_2, ..., w_n\}$, the text encoder learns word representations through an embedding layer and intermediately outputs hidden features $\mathbf{h}_w = \{h_{w_1}, h_{w_2}, ..., h_{w_n}\}$ by LSTM layers, as shown below:
$$h_{w_i} = TextEncoder(w_i, h_{w_{i-1}}) \tag{2}$$
- **Context Encoder:** The context encoder shares the same network architecture with the text encoder, i.e., input layer, embedding layer, and several LSTM layers. Similarly, the context encoder learns the high-level representations from the context sequence which is generated by the contextual information mapping strategy. The context information mapping strategy is described in Section 3.3. With a context sequence being $C = \{c_1, c_2, ..., c_m\}$, the context encoder learns the context representations $\mathbf{h}_c = \{h_{c_1}, h_{c_2}, ..., h_{c_m}\}$ by:
$$h_{c_i} = ContextEncoder(c_i, h_{c_{i-1}}) \tag{3}$$
- **Context-aware Decoder:** The context-aware decoder is constructed based on a context attention module and two FC layers. Specifically, the learned representations from the text encoder (AM output) and context encoder (contextual information) are fused with different weights optimized by the context attention module. Then, the first FC layer is applied to transform the fused features. The last FC layer with Softmax activation is applied to normalize the output probability on the vocabulary. The decoder process can be summarized as follows:
$$s_i = Score(h_{w_i}, \mathbf{h}_c) = V^T tanh(W h_{w_i} + U \mathbf{h}_c) \tag{4}$$
$$\alpha_i = softmax(s_i) = \frac{\exp(s_i)}{\sum_{j=1}^{N} \exp(s_j)} \tag{5}$$

$$c_i = \sum_{j=1}^{N} \alpha_i h_{c_j} \tag{6}$$

$$P(y_i | w_{<i}, C) = f\left(concat(h_{w_i}, c_i)\right) \tag{7}$$

The inference rule of the feature fusion method (context attention module) is motivation by the attention mechanism. Firstly, each hidden unit h_{w_i} from the text sequence is assigned the score s_i with the context vector $\mathbf{h}_c$ by Equation (4), where V, W, U are trainable parameters. Secondly, the scores $s_1, \ldots, s_i$ are normalized by the Softmax operation as in Equation (5) to get the fusion weights α_i. Then, in Equation (6), a weighted sum is calculated on the context feature c_i to obtain the fused context feature representation for step i. Finally, as shown in Equation (7), the text representation vector h_{w_i} and the fused context representation vector c_i of step i are concatenated to form a context-aware vector for the FC layer to generate an output y_i.

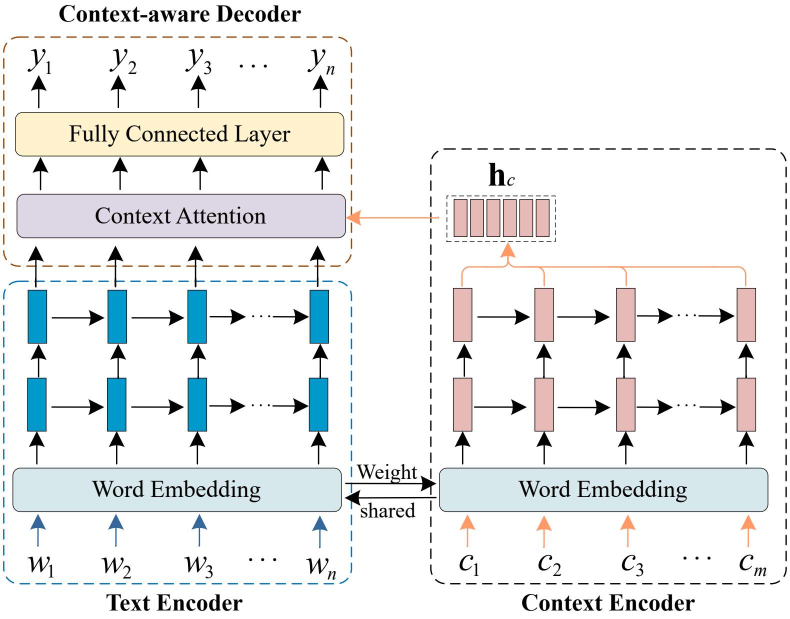


Figure 2. The architecture of the proposed CALM.

It is worth noting that the context sequence and text sequence share the same vocabulary. Meanwhile, the embedding layer of the context encoder and the text encoder also share the learned weights to build stronger correlations of the same vocabulary between the text sequence and context sequence. In this work, the architecture of the CALM is described as follows: the size of the embedding layer is set to 200 for both the text encoder and context encoder, followed by two LSTM layers with 200 neurons per layer. The context-aware decoder is configured with a context attention module and two fully connected layers with $|V|$ units (vocabulary size).

Finally, the proposed CALM is incorporated into the ASR system in two ways, i.e., decoding and rescoring. The decoding strategy is performed with a beam search algorithm (refer to to [35]). Beam search uses a breadth-first search strategy to build its search tree, which can easily integrate the scores of CALM into the search process. In the rescoring procedure, the N-best list of the AM decoding results by beam search is used as the candidate set to generate the final result.

3.3. Contextual Information Organization

In this paper, we mainly focus on integrating flight callsign knowledge into the ASR system to improve its performance in a real-time ATC environment. Basically, a total of two problems are required to be addressed for the context mapping:

- multiple pronunciations for a single callsign: the airline company name DLH can be spoken as “delta lima hotel” or Lufthansa. Similarly, the airline number “8883” can be spoken as "eight eight eight three" or "triple eight three" in English or "ba ba ba san" in Chinese.

- multiple callsign entities from the real-time context: in most cases, there are several flights in a control sector, which are required to be fed into the context encoder to support the subsequent decoding procedure.

In this work, the callsign of international flights is represented by the English word, while the Chinese character is for domestic flights. The rest of the contextual information is organized based on their standard pronunciation. By organizing contextual information in a different format, the CALM is expected to learn the inherited semantic representations of the same callsign entity.

Specifically, multiple flight callsigns are organized as a text sequence with a predefined separator, as: $callsign_1$ $<eos>$ $callsign_2$ $<eos>$ $callsign_3$ $<eos>, \dots, callsign_n$ $<eos>$, in which the $<eos>$ means the end of sentence and serves as a separator between callsign entities. Each callsign is regarded as a whole entity to provide discriminative features for different callsigns.

4. Experiments and Discussions

4.1. ATC Corpus

In this work, both the AM and LM are trained on the ATCSpeech corpus [15] that was collected from a real ATC environment. The ATCSpeech is a manually labeled multilingual ASR corpus, which includes about 39.83 h of Chinese speech and about 18.69 h of English speech. Moreover, this corpus covers all flight phases (ground, tower, approach, area control center) and is a more comprehensive ATC speech dataset. The detailed descriptions (i.e., duration, the number of utterances, speaker gender, and speaker role.) of the ATCSpeech corpus are described in Table 1. In addition, more details of the ATCSpeech corpus can be found in [15].

Table 1. The detailed descriptions of the ATC Speech corpus and the Test-real dataset.

Items		**ATCSpeech**		**Test-Real**	
		Chinese	**English**	**Chinese**	**English**
Amount	#Hours	39.83	18.69	3.56	1.46
	#Utterance	45,586	16,939	3411	1485
Speaker role (Hours)	Pilot	21.12	8.92	1.83	0.69
	Controller	18.73	9.77	1.73	0.77
Speaker gender (Hours)	Male	36.16	16.94	3.22	1.35
	Female	3.69	1.75	0.34	0.11

Since the contextual information of training samples in this dataset can no longer be traced back, a simulation strategy is applied to generate the input of the context encoder. To simulate the callsign for each utterance, the callsigns of the whole corpus are pre-extracted to formulate a callsign pool. About 4.5% of the samples are without a callsign in their transcription, which is labeled as *None*. In the training stage, the contextual information for each utterance is a combination set, including its own callsign and k randomly selected items from the callsign pool. Here, k is picked uniformly from [1, $N_{callsign}$], where $N_{callsign}$ is a hyperparameter of the training procedure.

To further validate the proposed approach, an extra test set (called test-real) is also organized to consider the influence of the simulated contextual situational information. The test-real was collected from the real ATC environment of Chengdu area control, including the ATC speech and real-time contextual situational information. The details of the test-real set are also shown in Table 1; there are 4896 utterances in this dataset with a total duration of about 5 h, about 70% Chinese speech, and 30% spoken in English.

4.2. Experimental Configurations

Due to the multilingual nature of the ATCSpeech corpus, three AM models, i.e., ASR-C, ASR-E, ASR-A, are applied to conduct experiments, as shown below:

- ASR-C: the model is optimized on the Chinese speech of the ATCSpeech corpus.
- ASR-E: the model is optimized on the English speech of the ATCSpeech corpus.
- ASR-A: the model is optimized on the whole ATCSpeech corpus.

Based on the above ASR models, the proposed CALM is evaluated on both the decoding and rescoring phases. In the decoding experiments, the output vocabulary of the CALM is the same as that of the ASR model, i.e., Chinese character and English letter for Chinese and English speech, respectively. To explore the effect of the modeling unit, both the English letter and word are regarded as the basic token to train the related LM for the N-best rescoring evaluation.

In addition, two comparative baselines, including the N-gram and RNNLM, are also designed to confirm the efficiency and effectiveness of the proposed approach. The N-gram LM is implemented based on the KenLM toolkit [37]. The order is set to 9 and 18 for Chinese and English speech, respectively, and 15 for multilingual speech. The RNNLM architecture is implemented by removing the context encoder and attention layer of the CALM, while other layers remain unchanged.

Based on the statistics of the real ATC environment, the hyperparameter $N_{callsign}$ is set to 20. In the test-real dataset, the number of callsigns depends on the collected real-time contextual information. The top-10 hypothesis of the decoding results is applied to achieve the rescoring procedure. The beam width of the decoding procedure is set to 20.

In this work, the character error rate (CER %) based on the Chinese character and English letter is applied to evaluate the ASR output, while the callsign accuracy (CSA %) is for the callsign identification task. Only when all elements in the callsign are correctly recognized can it be considered as a valid result.

The calculations of CER and CSA are shown as below:

$$CER = \frac{S+D+I}{N} \tag{8}$$

$$CSA = \frac{C_{callsigns}}{T_{utterances}} \tag{9}$$

where N is the length of the ground-truth, the S, D, and I are the number of the substitution, delete, and insert operations for converting the predicted label into the ground-truth. The $C_{callsign}$ and $T_{utterances}$ represent the number of utterances whose callsigns are correctly recognized and the total utterance in the test data set, respectively.

In the experiment, we construct and train all the models with the open-source deep learning framework PyTorch 1.4.0. The training server was configured as follows: Ubuntu 16.04 operating system with 2*NVIDIA GeForce RTX 2080Ti GPU, Intel Xeon E5-2630 CPU, and 128 GB memory. Cross-entropy is used as the loss function for both the RNNLM and CALM. The initial learning rate of the LM training process starts at 20 and anneals the learning rate (reduce to 1/4) if the validation loss had not improved at the end of every epoch.

4.3. Results

4.3.1. Decoding Results

The results of applying the proposed CALM to the decoding procedure are reported in Table 2. As can be seen from the results, an extra LM is able to significantly improve the ASR performance. Specifically, the N-gram and RNNLM correct some spelling errors of the AM outputs, and slightly improve the CSA. They can effectively correct the airline code, or the callsign has occurred in the training set. However, they fail to make positive contributions to correct unseen callsigns, especially isolated digits or letters in the callsign. It can be attributed that there are no semantic correlations between the digits or letters in the callsign. For both of the datasets, the proposed CALM achieves a considerable performance improvement for the callsign identification task, i.e., over 30% relatively CSA improvement for the ASR-C and ASR-A model, and about 73% for the ASR-E model.

Table 2. Results of the decoding procedure.

Methods		**Test Set**			
AM	**LM**	**Test**		**Test-Real**	
		CER%	**CSA%**	**CER%**	**CSA%**
ASR-C	-	8.10	65.58	7.94	64.07
	N-gram	6.31	74.17	6.49	71.88
	RNNLM	6.13	76.58	6.24	75.81
	CALM	**4.57**	**87.50**	**4.80**	**87.39**
ASR-E	-	10.40	46.54	10.81	45.68
	N-gram	9.20	62.88	9.35	61.79
	RNNLM	8.10	64.86	8.24	63.19
	CALM	**6.20**	**80.88**	**6.79**	**79.67**
ASR-A	-	6.96	65.34	7.35	66.17
	N-gram	5.95	73.57	6.37	71.74
	RNNLM	5.91	74.17	6.03	77.60
	CALM	**4.36**	**85.92**	**4.64**	**85.47**

It also can be seen from the experimental result that the models optimized on the whole corpus obtained better results than the ones optimized on the monolingual speech corpus. Firstly, the increase of training samples (the whole ATCSpeech corpus vs. Chinese or English speeches) helps to improve the performance of the model. Secondly, better performance of intra-sentential code-switching was presented on the Chinese-English speech in multilingual ASR systems.

Note that, since the decoding procedure is a sequential iterative search (no parallel computing), the computational complexity for the NN-based LM is much higher than that of the N-gram ones.

4.3.2. Rescoring Results

In this section, the proposed CALM is also applied to the ASR rescoring. Since the N-gram LM reaches a better trade-off between the performance and computational complexity, it serves as the LM for the decoding procedure (a baseline) in this section. Only the ASR-A model is used for this experiment due to its superior performance over independent systems. To further validate the LM modeling unit, both the English letter and word are applied to train the LM for the English speech, while the Chinese character is always for Chinese speech. The rescoring results for different modeling units are listed in Tables 3 and 4, respectively.

Table 3. The rescoring results with LM (English letter).

Methods		**Test Set**			
N-Best System	**LM**	**Test**		**Test-Real**	
		CER%	**CSA%**	**CER%**	**CSA%**
ASR-A	RNNLM	6.79	67.52	7.12	63.53
	CALM	**6.49**	**74.69**	**6.70**	**75.97**
ASR-A + N-gram	RNNLM	6.17	69.83	6.29	68.75
	CALM	**5.79**	**84.62**	**5.68**	**85.15**

Table 4. The rescoring results with LM (English word).

Methods		Test Set			
N-Best System	LM	Test		Test-Real	
		CER%	CSA%	CER%	CSA%
ASR-A	RNNLM	6.27	69.82	6.76	70.79
	CALM	**5.71**	**79.50**	**5.86**	**79.26**
ASR-A + N-gram	RNNLM	6.02	72.76	6.71	71.54
	CALM	**4.96**	**85.24**	**5.05**	**85.76**

The following conclusions can be drawn from the experimental results:

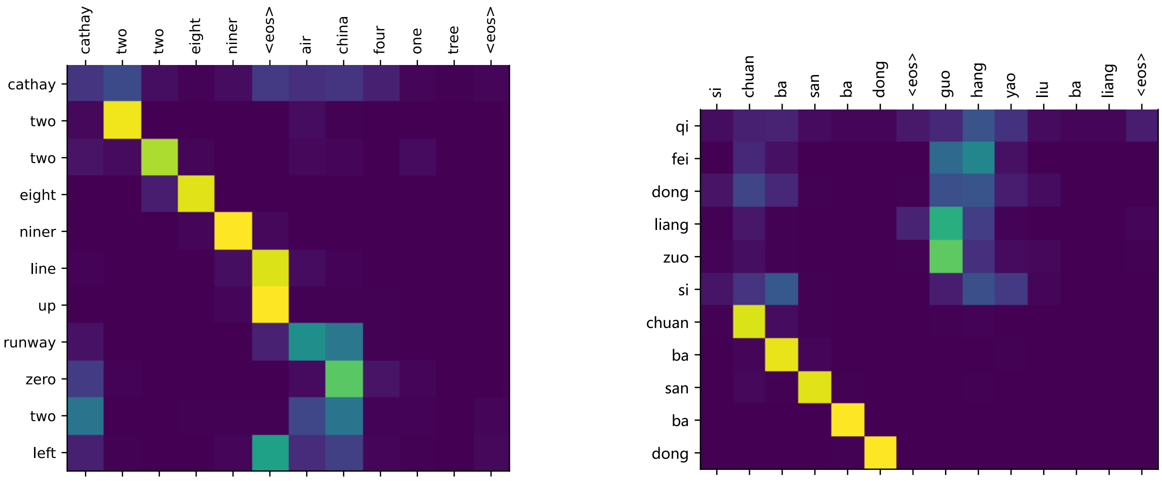
1. By using the N-gram LM for the decoding procedure, the final ASR performance is slightly improved with both the CALM and RNNLM rescoring. This fact also validates the decoding procedure, which provides a more reliable N-best list and further benefits to the rescoring procedure.
2. For the test and test-real datasets, the CALM outperforms the common LM for both the ASR and callsign identification task. Thanks to the contextual information, the CALM achieves about 85% CSA, i.e., 20% absolute improvement.
3. The LMs trained with English words obtain superior performance over those trained with English letters. It can be attributed that taking English letters as the modeling unit leads to the input sequence being too long to capture the vocabulary dependencies, which further affects the final performance of the NN-based LMs.
4. It can also be seen that, since the rescoring is a separate procedure without considering the AM probability, the rescoring results are not always optimal (the lowest CER) compared to that of applying it to the decoding procedure. However, the rescoring is a one-pass procedure, and can be achieved with less computational resources in a real-time manner. It is a more preferable way to take advantage of the proposed CALM in the real environment.

4.4. Visualization and Analysis

To better understand how the CALM works, the learned context attention weights are visualized in Figure 3 for both Chinese and English speech examples. The x-axis and y-axis correspond to the input of the context encoder (contextual information) and the text encoder (AM output), respectively. Purple colors denote the attention values close to 0, while the yellow colors represent the values close to 1. The outputs of the baselines and the CALM for given examples are also presented in Table 5.

Table 5. Comparing results output by the baselines and the proposed CALM.

Ground Truth	Contextual Information	Outputs			
		AM	KenLM	RNNLM	CALM
Cathay two two eight niner line up runway zero two left	*Cathay two two eight niner* <eos> Air China four ... United four five one <eos>	*Cathay two eight niner* line up runway zero two left	*Cathay two eight niner* line up runway zero two left	*Cathay two eight niner* line up runway zero two left	*Cathay two two eight niner* line up runway zero two left



(a) Attention weights of English speech.

(b) Attention weights of Chinese speech.

Figure 3. The learned context attention weights of the CALM in the English word based rescoring procedure. Note that the Chinese characters in the (**b**) are represented by Chinese pinyin.

As shown in Figure 3 and Table 5, compared to the baselines, the probabilities of the callsign were successfully biased by the proposed CALM, which properly considers the contextual information. In practice, the callsign *Cathay two eight niner* is also a valid expression in contextual-independent situations. Therefore, the conventional LM (i.e., baselines) failed to predict correct results. Thus, it is clear that the proposed CALM indeed captures the desired correlations between the contextual information and the AM output, which further supports the motivation of this work.

In practice, the requirements of the CER and CSA depend on the specific application scenario in the ATC. For instance, a lower CER (<5%) and a higher CSA (>85%) are needed to ensure accurate alarm in real-time speech understanding-based safety monitoring systems, while the 10% CER and 75% CSA are also acceptable in the speech data retrieval and analysis system. In summary, the proposed CALM was validated on the real-world dataset and can support the majority of ASR applications in the ATC domain.

5. Conclusions and Future Works

In this work, we propose to apply contextual information to improve the ASR performance in the ATC domain. To this end, a context-aware LM (based on an encoder–decoder architecture) is proposed to integrate predefined flight callsigns into the ASR system. By combining with an end-to-end ASR model, the proposed approach is validated on a multilingual real-world corpus. Experimental results show that it outperforms other baselines for both the ASR and callsign identification task, achieving 4.36% CER and about 85.92% CSA. Most importantly, the proposed approach is also confirmed in a real-time environment. Due to the computational complexity, we believe that the ASR rescoring is a preferable way to practically take advantage of the proposed approach.

In the future, we plan to integrate more situational context information (such as speed, altitude.) into the proposed CALM to improve the performance of recognizing the key ATC elements in the ASR system.

Author Contributions: Methodology, D.G., Z.Z. and B.Y.; validation, Z.Z.; data curation, Z.Z.; software, P.F.; investigates, P.F.; writing—original draft preparation, D.G.; writing—review and editing, J.Z. and B.Y.; supervision, B.Y.; project administration, J.Z.; funding acquisition, J.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China, Grant Nos. 62001315 and U20A20161.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data sharing not applicable.

Acknowledgments: The authors would like to thank all contributors to the ATCSpeech corpus.

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

References

1. Ohneiser, O.; Helmke, H.; Ehr, H.; Gürlük, H.; Hössl, M.; Kleinert, M.; Mühlhausen, T.; Uebbing-Rumke, M.; Oualil, Y.; Schulder, M.; et al. Air Traffic Controller Support by Speech Recognition. In Proceedings of the International Conference on Applied Human Factors and Ergonomics (AHFE), Krakow, Poland, 19–23 July 2014; pp. 492–503.
2. 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]
3. Lin, Y.; Deng, L.; Chen, Z.; Wu, X.; Zhang, J.; Yang, B. A Real-Time ATC Safety Monitoring Framework Using a Deep Learning Approach. *IEEE Trans. Intell. Transp. Syst.* **2020**, *21*, 4572–4581. [CrossRef]
4. Ferreiros, J.; Pardo, J.; de Córdoba, R.; Macias-Guarasa, J.; Montero, J.; Fernández, F.; Sama, V.; d'Haro, L.; González, G. A speech interface for air traffic control terminals. *Aerosp. Sci. Technol.* **2012**, *21*, 7–15. [CrossRef]
5. Lin, Y.; Wu, Y.; Guo, D.; Zhang, P.; Yin, C.; Yang, B.; Zhang, J. A Deep Learning Framework of Autonomous Pilot Agent for Air Traffic Controller Training. *IEEE Trans. Hum. Mach. Syst.* **2021**, *51*, 442–450. [CrossRef]
6. Zuluaga-Gomez, J.; Vesel, K.; Blatt, A.; Motlicek, P.; Landis, F. Automatic Call Sign Detection: Matching Air Surveillance Data with Air Traffic Spoken Communications. *Proceedings* **2020**, *59*, 14. [CrossRef]
7. Nguyen, V.N.; Holone, H. Possibilities, Challenges and the State of the Art of Automatic Speech Recognition in Air Traffic Control. *Int. J. Comput. Inf. Eng.* **2015**, *9*, 1933–1942.
8. Lin, Y. Spoken Instruction Understanding in Air Traffic Control: Challenge, Technique, and Application. *Aerospace* **2021**, *8*, 65. [CrossRef]
9. Lin, Y.; Guo, D.; Zhang, J.; Chen, Z.; Yang, B. A Unified Framework for Multilingual Speech Recognition in Air Traffic Control Systems. *IEEE Trans. Neural Networks Learn. Syst.* **2021**, *32*, 3608–3620. [CrossRef]
10. Zuluaga-Gomez, J.; Motlicek, P.; Zhan, Q.; Veselý, K.; Braun, R. Automatic Speech Recognition Benchmark for Air-Traffic Communications. *Proc. Interspeech* **2020**, *2020*, 2297–2301. [CrossRef]
11. Srinivasamurthy, A.; Motlícek, P.; Himawan, I.; Szaszák, G.; Oualil, Y.; Helmke, H. Semi-Supervised Learning with Semantic Knowledge Extraction for Improved Speech Recognition in Air Traffic Control. In Proceedings of the 18th Annual Conference of the International Speech Communication Association, Stockholm, Sweden, 20–24 August 2017; pp. 2406–2410.
12. 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]
13. Lin, Y.; Li, Q.; Yang, B.; Yan, Z.; Tan, H.; Chen, Z. Improving speech recognition models with small samples for air traffic control systems. *Neurocomputing* **2021**, *445*, 287–297. [CrossRef]
14. Pellegrini, T.; Farinas, J.; Delpech, E.; Lancelot, F. The Airbus Air Traffic Control Speech Recognition 2018 Challenge: Towards ATC Automatic Transcription and Call Sign Detection. *Proc. Interspeech* **2019**, *2019*, 2993–2997. [CrossRef]
15. Yang, B.; Tan, X.; Chen, Z.; Wang, B.; Ruan, M.; Li, D.; Yang, Z.; Wu, X.; Lin, Y. ATCSpeech: A Multilingual Pilot-Controller Speech Corpus from Real Air Traffic Control Environment. *Proc. Interspeech* **2020**, *2020*, 399–403. [CrossRef]
16. Delpech, E.; Laignelet, M.; Pimm, C.; Raynal, C.; Trzos, M.; Arnold, A.; Pronto, D. A Real-life, French-accented Corpus of Air Traffic Control Communications. In Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018), Miyazaki, Japan, 7–12 May 2018.
17. Mohri, M.; Pereira, F.; Riley, M. Speech Recognition with Weighted Finite-State Transducers. In *Springer Handbook of Speech Processing*; Benesty, J., Sondhi, M.M., Huang, Y.A., Eds.; Springer: Berlin/Heidelberg, Germany, 2008; pp. 559–584. [CrossRef]
18. Novak, J.R.; Minematsu, N.; Hirose, K. Dynamic Grammars with Lookahead Composition for WFST-based Speech Recognition. *Proc. Interspeech* **2012**, *2012*, 1079–1082.
19. Hall, K.B.; Cho, E.; Allauzen, C.; Beaufays, F.; Coccaro, N.; Nakajima, K.; Riley, M.; Roark, B.; Rybach, D.; Zhang, L. Composition-based on-the-fly rescoring for salient n-gram biasing. In Proceedings of the 16th Annual Conference of the International Speech Communication Association, Dresden, Germany, 6–10 September 2015; pp. 1418–1422.
20. Aleksic, P.S.; Ghodsi, M.; Michaely, A.H.; Allauzen, C.; Hall, K.B.; Roark, B.; Rybach, D.; Moreno, P.J. Bringing contextual information to google speech recognition. *Proc. Interspeech* **2015**, *2015*, 468–472. [CrossRef]
21. Huang, R.; Abdel-hamid, O.; Li, X.; Evermann, G. Class LM and Word Mapping for Contextual Biasing in End-to-End ASR. *Proc. Interspeech* **2020**, *2020*, 4348–4351. [CrossRef]
22. Chan, W.; Jaitly, N.; Le, Q.; Vinyals, O. Listen, attend and spell: A neural network for large vocabulary conversational speech recognition. In Proceedings of the 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Shanghai, China, 20–25 March 2016; pp. 4960–4964. [CrossRef]
23. Zhao, D.; Sainath, T.N.; Rybach, D.; Rondon, P.; Bhatia, D.; Li, B.; Pang, R. Shallow-Fusion End-to-End Contextual Biasing. *Proc. Interspeech* **2019**, *2019*, 1418–1422. [CrossRef]
24. Graves, A. Sequence Transduction with Recurrent Neural Networks. *arXiv* **2012**, arXiv:1211.3711

25. Pundak, G.; Sainath, T.N.; Prabhavalkar, R.; Kannan, A.; Zhao, D. Deep Context: End-to-end Contextual Speech Recognition. In Proceedings of the 2018 IEEE Spoken Language Technology Workshop, SLT 2018, Athens, Greece, 18–21 December 2018; pp. 418–425. [CrossRef]
26. Jain, M.; Keren, G.; Mahadeokar, J.; Zweig, G.; Metze, F.; Saraf, Y. Contextual RNN-T for Open Domain ASR. *Proc. Interspeech* **2020**, *2020*, 11–15. [CrossRef]
27. Oualil, Y.; Klakow, D.; Szaszák, G.; Srinivasamurthy, A.; Helmke, H.; Motlícek, P. A context-aware speech recognition and understanding system for air traffic control domain. In Proceedings of the 2017 IEEE Automatic Speech Recognition and Understanding Workshop, Okinawa, Japan, 16–20 December 2017; pp. 404–408. [CrossRef]
28. Zuluaga-Gomez, J.; Nigmatulina, I.; Prasad, A.; Motlícek, P.; Veselý, K.; Kocour, M.; Szöke, I. Contextual Semi-Supervised Learning: An Approach To Leverage Air-Surveillance and Untranscribed ATC Data in ASR Systems. *arXiv* **2021**, arXiv:2104.03643.
29. Shore, T.; Faubel, F.; Helmke, H.; Klakow, D. Knowledge-Based Word Lattice Rescoring in a Dynamic Context. *Proc. Interspeech* **2012**, *2012*, 1083–1086.
30. Schmidt, A.; Oualil, Y.; Ohneiser, O.; Kleinert, M.; Schulder, M.; Khan, A.; Helmke, H.; Klakow, D. Context-based recognition network adaptation for improving online ASR in Air Traffic Control. In Proceedings of the 2014 IEEE Spoken Language Technology Workshop, SLT 2014, South Lake Tahoe, NV, USA, 7–10 December 2014; pp. 13–18. [CrossRef]
31. Oualil, Y.; Schulder, M.; Helmke, H.; Schmidt, A.; Klakow, D. Real-time integration of dynamic context information for improving automatic speech recognition. *Proc. Interspeech* **2015**, *2015*, 2107–2111.
32. 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.
33. Sriram, A.; Jun, H.; Satheesh, S.; Coates, A. Cold fusion: Training Seq2seq Models Together with Language Models. In Proceedings of the 6th International Conference on Learning Representations, Vancouver, BC, Canada, 30 April–3 May 2018.
34. Lin, Y.; Yang, B.; Guo, D.; Fan, P. Towards multilingual end-to-end speech recognition for air traffic control. *IET Intell. Transp. Syst.* **2021**, *15*, 1203–1214. [CrossRef]
35. Amodei, D.; Ananthanarayanan, S.; Anubhai, R.; Bai, J.; Battenberg, E.; Case, C.; Casper, J.; Catanzaro, B.; Chen, J.; Chrzanowski, M.; et al. Deep Speech 2: End-to-End Speech Recognition in English and Mandarin. In Proceedings of the 33nd International Conference on Machine Learning, New York, NY, USA, 19–24 June 2016; Volume 48, pp. 173–182.
36. Graves, A.; Fernández, S.; Gomez, F.; Schmidhuber, J. Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks. In Proceedings of the 23rd International Conference on Machine Learning—ICML 2006, Pittsburgh, PA, USA, 25–29 June 2006; pp. 369–376. [CrossRef]
37. Heafield, K.; Lavie, A. Combining Machine Translation Output with Open Source: The Carnegie Mellon Multi-Engine Machine Translation Scheme. *Prague Bull. Math. Linguist.* **2010**, *93*, 27–36. [CrossRef]