Frontiers in Computing and Intelligent Systems
ISSN: 2832-6024 | Vol. 8, No. 2, 2024

End-to-End Speech Hash Retrieval Algorithm based on Speech Content and Pre-training

Yian Li *, Yibo Huang

College of Physics and Electronic Engineering, Northwest Normal University, Lanzhou Gansu, 730070, China
* **Corresponding author**: Yian Li (Email: 15161122091@163.com)

Abstract: Traditional speech retrieval tasks, such as audio fingerprinting and Spoken Word Detection Query (STD-QbE), focus on feature matching for speech or keyword retrieval. In this paper, we present a content-based speech retrieval algorithm. The algorithm allows matching based on the complete content of sentences in speech, not just local features or keywords. Importantly, it completely bypasses the Automatic Speech Recognition (ASR) transcription process by mapping the acoustic features of the sentence directly to the Hamming space. Retrieval of the same content is then achieved by comparing Hamming distances, thus effectively eliminating the potential impact of transcription errors on retrieval performance. In order to achieve this, our approach employs the Connectionist Temporal Classification (CTC) speech recognition technique to pre-train the model to learn content-dependent representations of speech features. Through experiments, we demonstrate that our approach achieves excellent performance in speech retrieval tasks.

Keywords: Speech Retrieval; Deep Hashing; WaveNet; Transformer; Pre-training.

1. Introduction

Speech is gaining prominence in today's society as one of the most relied upon forms of expression in modern times. Written modes of communication, such as electronic and text messages, may not convey information with complete accuracy because humans may misunderstand them. Textual modes of communication may not be able to convey information with complete accuracy because humans may misunderstand them. Speech, on the other hand, does not have this problem [1][2]. In addition, with addition, voice interaction is becoming more and more common with the popularity of mobile and smart home devices, making speech technology more and more feasible. Content-based speech retrieval allows users to search for similar or matching content directly in large speech databases using speech segments as query criteria. The advantage of this approach is that the user does not need to know the exact user only needs to provide a speech sample to complete the search. retrieval. This technique is particularly suitable for speech-independent scenarios such as human-machine speech interaction. text-independent scenarios such as machine-speech interaction. Although some progress has been made in speech retrieval technology, there are still some unsolved problems and challenges in practical applications.

Mainstream content-based speech retrieval algorithms generally rely on speech recognition technology. This technology firstly needs to convert speech content into text form, and then utilize mature text retrieval techniques to realize effective retrieval of information. [3-6] This approach not only allows for precise search based on the specific content of speech, but also allows for deeper exploration of its underlying semantic information. Although the use of transcription technology has largely improved the accuracy of speech retrieval, its efficiency and accuracy are highly dependent on advanced speech recognition technology. Building reliable speech recognition systems requires large transcription datasets and language-specific expertise. However, collecting such a large and accurate dataset is a daunting task for many application scenarios. In the absence of sufficiently labeled data, speech recognition systems often struggle to fully grasp domain-specific terminology or linguistic properties, which directly affects the accuracy and reliability of their application within the speech retrieval domain.

Especially in the context of the era of data explosion, content-based speech retrieval faces another major challenge - the efficiency problem. [7-9] With the rapid growth in the volume of speech data, the traditional approach of traversing the entire database to match the query data becomes extremely time-consuming and inefficient. The limitations of this approach are especially prominent when dealing with large-scale datasets.

To address this problem, we introduce a pre-training-based end-to-end speech retrieval strategy focusing on acoustic-level sentence matching. Unlike traditional retrieval methods, our approach aims to capture and compare the complete sentence content of the entire speech segment. In order to achieve fast retrieval, the algorithm also maps the entire speech sentence content into equal-length sequences in Hamming space, constructing a hash index table for fast matching in the retrieval task. The main contributions of this paper can be summarized as follows:

1.To circumvent the transcription problem inherent in traditional cascade speech retrieval, we introduce an end-to-end acoustic-level content matching strategy. For this purpose, we use WaveNet as the encoder of the model and pre-train it with CTC speech recognition to enable it to pre-extract content-related information from the speech signal.

2. In order to achieve efficient content retrieval in large speech databases, we employ deep hashing techniques. Relying on the powerful characterization capability of deep learning, by projecting the whole speech content directly into to the Hamming space and comparing the Hamming distances, our method significantly improves the retrieval efficiency compared to direct speech matching.

The rest of the paper is organized as follows: in Section 2, we introduce the technical issues related to the algorithm.

Section 3 provides a detailed description of the algorithm. In Section 4 we present the experimental dataset and setup. Finally, Section 5 summarizes the proposed work.

2. Related Works

2.1. Deep Hashing

In the task of nearest neighbor search, hashing methods are widely adopted for their excellent computational and storage efficiency. The goal is to convert high-dimensional feature vectors into binary hash codes, ensuring that the hash codes of similar data are as close as possible. Traditional hashing algorithms usually consist of two parts: a mapping function to handle the mapping of the input data to the hash code, and a key-value storage table to store the hash code for easy lookup. In the field of speech retrieval, perceptual hashing techniques are widely used. The core idea of perceptual hashing is to capture the perceptual features of the speech signal and encode them into a binary hash code, whose hash function is the perceptual features of the speech (timbre, prosody, resonance features, etc.). However, since perceptual hashing mainly focuses on the low-level perceptual features of speech rather than its high-level semantic content, it is not suitable for applications that require in-depth understanding of speech content. To solve this problem, deep hashing techniques, which apply deep learning to hashing algorithms, are beginning to receive attention from researchers.

Deep neural networks can learn a set of hash functions customized for a specific task. Ideally, a deep hash network preserves semantic similarity information of multimedia data during hash construction.

Deep hashing has higher accuracy, better generalization ability and stronger robustness in image retrieval than traditional feature-based manual hashing methods [10-13]. With the successful application of deep hashing in the image domain, researchers have begun to explore its potential in speech retrieval. ZHANG et al. used neural networks to learn depth-aware features and generate depth-aware hash sequences to achieve efficient speech retrieval with good discriminative and robustness [14].Yuan et al. used deep hash code embedding instead of a large number of AWEs to achieve keyword-based of fast QbE speech retrieval [15]. Our content-based QbE speech retrieval algorithm is similar to the sentence embedding technique that transforms sentences or text passages into fixed-size vectors by mapping the contents of speech sentences into Hamming space and matching them by a normalized Hamming distance algorithm.

2.2. Content-based Speech Retrieval

The objective of the speech retrieval task is to identify speeches with content identical to a given speech sample. Traditional methods[16] primarily rely on ASR technology to transcribe speech into text, followed by comparative retrieval. This approach typically involves two primary steps: initially, the entire speech database is transcribed into text using ASR technology; subsequently, text retrieval techniques are employed to search for specific words or phrases within the transcribed text. A significant advantage of this method is its ability to swiftly search within the speech database. However, a pronounced drawback is its total dependence on the accuracy of ASR. Consequently, if there are errors in the ASR transcription or if the queried vocabulary is absent from the ASR lexicon (i.e., OOV terms), the retrieval performance might be severely compromised. To address the matching challenge at the acoustic level, researchers introduced the DTW method. For instance, in the research by Muaidi et al.[17], DTW was utilized for Arabic audio news retrieval.

DTW allows a nonlinear mapping between two speech signals to find the minimum distance between them. Although the DTW method has facilitated end-to-end speech retrieval, it is not without its challenges, such as high computational complexity, reduced efficiency in processing lengthy speech segments, and the potential to overlook latent linguistic details. With the emergence of deep learning technologies, researchers have been exploring avenues to harness deep learning to surmount the limitations of these conventional methods. Acoustic Word AWEs, for instance, represent an application of deep learning in speech retrieval. The modus operandi of AWEs is to employ deep neural networks to learn representations of speech segments. These networks are trained to extract meaningful features from the raw speech signal and map these features to a low-dimensional space, typically a fixed-length vector. These vectors, or ”embeddings,” serve as a compact representation of speech segments, retaining crucial information from the original speech while eliminating irrelevant or redundant content. As demonstrated in the research by Shen et al.[18], they proposed a novel and effective approach by jointly training acoustic phoneme and word embed dings for end-to-end Text-to-Speech(TTS) systems. Contrasting with AWEs, which primarily focuses on keywords, our method in this paper accentuates the comparison of entire sentences, achieving QbE speech retrieval with a more comprehensive content match

2.3. Causal Dilated Convolution and WaveNet

WaveNet is a speech generation model proposed by DeepMind in 2016, which is able to model the raw speech data directly, with the predicted distribution of each audio sample conditional on all previous samples[19]. It can generate natural speech signals on text-to-speech tasks. The concepts of causal convolution and dilated convolution are proposed in WaveNet. Causal convolution does not depend on any future time step when processing the current time step[20]. Since models with causal convolution do not have recursive connections, they are usually trained faster than recurrent neural networks (RNNs), and their fast training speed becomes more pronounced when the sequence length is longer. Dilated convolution improves the efficiency of applying filters over large regions by skipping parts of the input data in certain steps. Although WaveNet is designed as a generative model, it can be directly applied to speech recognition tasks. Gao et al. presented the addition of local attention to WaveNet-CTC to increase the performance of Tibetan language recognitionin multi-task learning.[21].

2.4. Multihead Self-Attention Mechanism and Transformer

Transformer, an E2E model proposed by Google in 2017, is a model in which both encoders and decoders rely on self-attention to compute their inputs and outputs, and it is the first transduction model that does not use sequence-aligned RNNs or CNNs, which have received a lot of attention for their effectiveness in areas such as computer vision and natural language processing. A transformer is a multilayer architecture formed by transformer blocks stacked together. The transformer block consists of a multi-headed self-attentive mechanism, a position feedforward network, a layer

normalization module, and a residual joint.

The self-attention mechanism is the key concept of Transformer. Compared with RNN and LSTM, the self-attentive mechanism is more likely to capture long-distance interdependent features in a sequence. It represents the connection between any two steps in the sequence directly by a computational result, and the distance between distant-dependent features is greatly reduced, which facilitates the effective use of these features. The multi-headed attention mechanism consists of multiple self-attentions that can simultaneously attend to information from different representation subspaces at different locations. Transformer and its derived models have been widely used in tasks such as audio classification [22], sentiment analysis [23], etc. Organization of the Text

3. The Method

3.1. Model

In order to realize speech retrieval through speech content, we design a hybrid model. The model maps the distances between the contents to the Hamming space, generates binary hash codes related to the contents, and completes the matching by comparing the Hamming distances between the hash codes. The model consists of three main parts: a WaveNet-based encoder, a downsampling module based on multilayer causal convolution and pooling, and a decoder consisting of a transformer block. Each module is described in detail below.

Pre-training encoder: In order to extract content-related features from the speech signal, we employ a WaveNet-based pre-training encoder. The construction of the encoder starts with the conversion of the MFCC features of the speech into a set of initial features. This process is achieved by applying a one-dimensional convolutional process to the input data, followed closely by the batch normalization and the application of an activation function to enhance the model's nonlinear processing capability . Next, the encoder introduces multiple residual blocks that are arranged repeatedly with different expansion rates to cover different time spans. Each residual block underwent feature extraction and generated a residual output for subsequent processing. In this way, the encoder is able to synthesize information from each time step, enhancing the understanding of the speech signal. Finally, the encoder combines all residual outputs and further processes them through another layer of activation functions to obtain a comprehensive feature representation. During pre-training, a layer of convolution is added at the end of the encoder for mapping the extracted high-level features to a specific output space, the size of which corresponds to the size of the vocabulary in the speech recognition task. In short, the purpose of this layer is to convert the complex speech features learned by the deep network into specific vocabulary predictions. The encoder is pre-trained with the CTC loss function.

Downsampling module: when processing acoustic features extracted by WaveNet pre-trained encoders, we noticed that the contents in neighboring time steps are often highly similar or even identical, which leads to data redundancy. In order to deal with this problem effectively and reduce the computational burden in subsequent processing steps, this study introduces causal convolution and pooling layers for data downsampling and feature screening. The design of this approach is inspired by classical convolutional neural network architectures, such as VGG and AlexNet, which gradually extract more advanced feature representations by cascading convolutional and pooling layers. In particular, with reference to VGG and AlexNet's strategy of increasing the number of filters with network depth, the model additionally adds a fixed number of filters at each additional layer. In this way, we retain the key information of the acoustic features while reducing the computational pressure for subsequent tasks through effective data dimensionality reduction. The network layer stacking begins with a convolutional layer, which performs feature extraction on the input speech data, and these features are subsequently fed into a normalization layer to stabilize the training process and improve generalization. Immediately after, the activation function layer introduces nonlinearities that allow the model to capture more complex data patterns. Finally, the pooling layer downsamples the activated features, effectively reducing the data dimensionality while enhancing the robustness of the model to input variations.

Decoder: In order to retrieve the processed sequence signals, this paper employs a series of transformer blocks as sequence matching decoders. transformer blocks and their derived models have been used in sequence classification problems, and models like BERT have achieved commendable results in text classification tasks. When dealing with text problems, the BERT model introduces a special classification Token ([CLS]) at the beginning of the sequence. Borrowing from this design, we add a vector with all values set to 1 in front of the processed sequence as the retrieval Token for this task. its final hidden state is considered as the retrieval feature and the deep hash feature i.e., the output of the model. The entire sequence matching decoder consists of multiple layers of transformer blocks. In order to enable the sequence matching decoder to learn the time dependent features, we introduce position encoding before feeding the sequence into the retrieval module. Position encoding helps the attention mechanism to understand the positional relationships within the sequence, which is crucial for sequence retrieval. The tanh activation function is employed to process the final hidden state of the retrieved Token. tanh activation function is formulated: $tanh(x)=\frac{e^{x}+e^{-x}}{e^{x}-e^{-x}}$, which is capable of restricting the output values to the range (-1,1), is mainly intended to facilitate the binarization of the model's output into hash codes.

3.2. Model Training

A two-stage training strategy was used for model training. In the first stage, the WaveNet encoder part is pre-trained. The second segment trains the complete model for classification based on speech content.

Pre-training phase: first, at the backend of the encoder, we added a one-dimensional convolutional layer aimed at obtaining the word probability distribution for each frame. Subsequently, training is performed using this newly added model structure with a Connected Timing Classification (CTC) loss function. Pre-training was performed using the same dataset as the classification training performed later.

Complete model training: our training goal is to ensure that the model is able to generate accurate hash codes for specific speech segments. To achieve this goal, a classification task is used as the main training strategy. Specifically, speech is categorized based on the textual information of the speech content, and the same textual information uttered by different

individuals is considered as the same category. In the model architecture, a fully-connected layer is added to the back-end of the model to accommodate the classification task. This layer is designed so that the model can output a vector containing the probability of each category. Through the application of a softmax activation function, these outputs are converted into probability distributions where each element represents the predicted probability of the corresponding category. Immediately thereafter, the cross-entropy loss function is utilized to evaluate the deviation between the probability distribution predicted by the model and the true labels, an approach that accurately quantifies the performance of the model on the classification task. Notably, despite starting with a pre-trained WaveNet model, the weights of the pre-trained portion are not frozen during subsequent training. Instead, the entire model is fine-tuned to ensure optimal performance on specific speech retrieval tasks. This fine-tuning strategy allows the model to combine the knowledge learned from the original pre-training task with the data from the new task to achieve better generalization performance. The details of the model are shown in Figure 1.

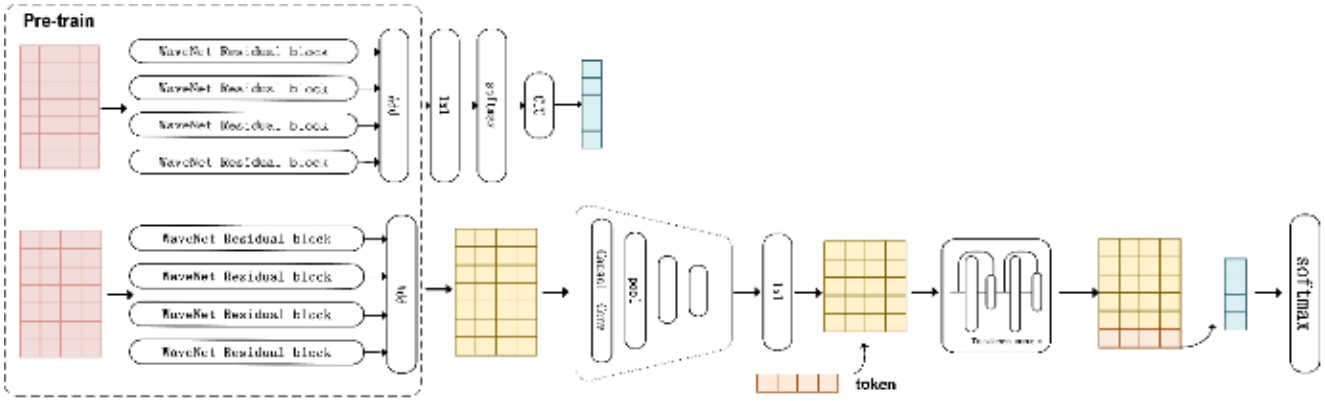

Figure 1. The Model

3.3. Hash Construction

To generate the hash code, the model passes the speech fragment through the model through forward propagation to the output. This output is a continuous value whose range lies between [-1,1]. To convert these continuous values into binary hash codes, we employ a simple thresholding strategy: for values greater than 0, we convert them to 1; for values less than or equal to 0, we convert them to 0. Specifically, given the model's modeled output y for a given speech fragment, its corresponding binary hash code b can be obtained as follows:

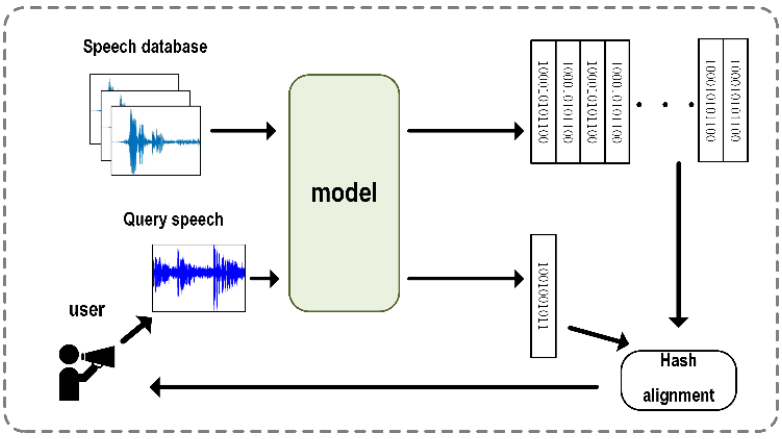

Figure 2. Retrieval Strategy

$$b_i = \begin{cases} 1 & \text{if } y_i > 0 \\ 0 & otherwise \end{cases}$$

3.4. Retrieval Strategy

This study proposes an end-to-end framework that accurately maps text-related content in speech to binary hash codes through deep learning techniques. The core of the approach is that it not only learns the mapping relationship from speech signals to textual content, but is also able to convert this mapping relationship directly into binary hash codes.

Before performing the retrieval task, the speech data in the database is preprocessed to generate a hash index table. In the retrieval phase, for a target speech, its binary hash code is first generated by the above model , and then this hash code is compared with the pre-stored hash code index table in the database. By calculating the bit error rate (BER), We can quickly identify the speech segments that are most similar in content to the target speech. This approach makes it possible to perform content-similar speech retrieval with great efficiency and accuracy even in huge speech databases, substantially improving the performance of speech retrieval systems. The retrieval framework is shown in Figure 2

4. Experiment

4.1. Experimental Setup

4.1.1. Database

The experiments in this thesis use the THCHS30 Chinese

speech database [24], whose selected speech content is derived from a large number of news sources. Based on the speech content, the dataset was divided into four groups, A, B, C, and D. The combination of words in groups A, B, and C served as a training set containing 30 speakers, while group D served as a test set containing 10 speakers. Speech files with the same ID in each group have the same content, each speech has a different duration, and each speech has not only text labels but also phoneme labels. We selected 6000 voices from group ABC as the training set, and 1000 voices from group D with 100 different contents as the test set. The labels are the phoneme labels of each speech in the dataset for pre-training, and the classification labels (one-hot codes) generated from the textual content of the speech for training.

4.1.2. Model Setup

1. Encoder: 15 WaveNet residual blocks were used to stack the WaveNet residual blocks with causal null convolution having 128 filters of filter size 7 and null rates of 1, 2, 4, 8 and 16 looped three times from input to output. The input data is processed simultaneously in the residual block by two convolutional layers, one of which uses a tanh activation function and the other uses a sigmoid activation function, and the outputs of these two convolutional layers are combined by element-by-element multiplication. This is the gated linear unit, the output of the gated linear unit is processed by both the convolutional layers, the tanh activation function, the output of one of them is used to pass on to the next residual block and the output of the other one is output after connecting it with the input residuals and is used for the subsequent residuals to be connected. The encoder will sum all the residual block outputs by element to encoder output.

2. Downsampling module: the entire downsampling module contains five blocks, each of which includes causal convolution, batch normalization, ReLU activation function, and maximum pooling. The filter size of the convolution layer is 15 and the number of filters is (L+1) × 128, where L represents the layer number. The pooling layer has a width of 2, making the data time step for each layer halved.

3. Decoder: The positional encoding is added before the data enters the decoder. The number of headers in the two-layer Transformer block structure after the decoder is set to 24. The output of the decoder is obtained from the final hidden state of the Token in the data through the tanh activation function.

4.1.3. Training Environment Setup

Pre-training is done using SGD (stochastic gradient descent) optimizer, the learning rate is initially 0.01, the data is divided into training set and validation set by 9:1, the learning rate decreases to 20% of the original in the next epoch when the loss value doesn't decrease, and the early stopping condition is set to the validation set loss of 5 epochs without decreasing. The encoder weights for classification training are derived from the pre-training, while the other parts of the initial weights are randomly set between -0.05 and 0.05. The model is trained using Adam optimizer with a learning rate of 0.001. All experimental results are based on 50 training cycles and the training data is divided into training and validation sets in a ratio of 80:20. The platform configuration used for all training is: intel core i5-12400, RTX3060 12G, python 3.9, tensorflow 2.6.0.

4.2. Comparison with Traditional Models

In order to show that the hybrid model in this paper has a significant advantage in end-to-end speech content hash matching, a bi-directional LSTM model is chosen as the control group, which has long been popular in the field of speech recognition and also used in the task of speech hash extraction. The control model has the same amount of data as the model in this paper. The specific control data is shown in Table 1.

Table 1. Comparison with baseline model

Model	Retrieval of indicators				
	mAP	MRR	P@1	P@5	P@9
this method	0.912	0.984	0.993	0.981	0.919
Bi-LSTM	0.310	0.497	0.645	0.488	0.380

Based on the above table, it can be seen that there is a huge gap between the traditional model applied directly on this task and the hybrid model of this study, which proves the superior performance of the hybrid model of this study on this task.

4.3. The Effect of Noise Interference on Speech Retrieval

In practical situations, speech retrieval tasks may be affected by noise interference or channel interference, which can lead to erroneous retrieval results. Therefore, in this paper, we investigate the robustness of the method to noise interference in speech retrieval tasks. In order to be able to perform speech retrieval in noisy environments, we manually added Gaussian white noise to the test dataset and constructed speech retrieval tasks with different signal-to-noise ratios (SNRs), including 30 db, 25 db, 20 db, 15 db, and 10 db. In order to simulate the training-testing mismatch, we trained the model by using the speech files with no added noise as the training set. Table 2 lists the speech retrieval performance of this method under different noise conditions. Unsurprisingly, the speech retrieval performance gradually decreases as the noise increases, but even at 10db, the present method still maintains a certain retrieval performance and is able to approach the situation when no noise is added at 30db. This proves the robustness of our method to noise interference.

Table 2. Retrieval performance in different noise environments

retrieval environment	Retrieval of indicators				
	mAP	MRR	P@1	P@5	P@9
Clean	0.912	0.984	0.993	0.981	0.919
SNR=30db	0.817	0.926	0.966	0.936	0.833
SNR=25db	0.815	0.939	0.969	0.937	0.831
SNR=20db	0.755	0.871	0.957	0.908	0.775
SNR=15db	0.651	0.791	0.927	0.845	0.684
SNR=10db	0.473	0.636	0.860	0.689	0.526

4.4. Impact of Pretraining Strategies on Speech Retrieval

In this section, the effectiveness of the pre-trained fine-tuned encoder will be verified by the present method and the following two comparative systems.

1.This method: the weight values of the encoder module are obtained by pre-training, and the weight values will not be frozen during the classification training process.

2.Pre-training and frozen encoder weights: the weight values of the encoder module are obtained by pre-training, and the weight values are frozen during the classification training.

3. No pre-training: direct classification training without

pre-training

Table 3. Impact of Pretraining Strategies on Retrieval Performance

Pre-training strategies	Retrieval of indicators				
	mAP	MRR	P@1	P@5	P@9
Pre-training and fine-tuning	0.912	0.984	0.993	0.981	0.919
Pre-trained with frozen weights	0.802	0.895	0.970	0.932	0.821
No pre-training	0.786	0.903	0.956	0.918	0.806

The experimental results are shown in Table 3, where pre-training the encoder can enhance significantly improve the performance of the model. In addition, using a fine-tuning strategy can yield better results than freezing the layer weights. Pre-training can help the model to learn features that are relevant to the speech content and the simple retrieval training may not be effective in updating the deeper parameters in the network, which can be alleviated by pre-training. A fine-tuning strategy allows the model to be personalized for a specific task while maintaining the knowledge learned by the pretrained model, allowing the model to adjust its parameters more finely to optimize performance for a specific task.

4.5. Impact of Pre-trained Labels on Speech Retrieval

This section analyzes the effect of pre-trained labels on speech retrieval by using Chinese characters as pre-trained labels in comparison to phonemes as labels for pre-training.

Table 4. Impact of pre-trained labels on retrieval performance

Pre-training labels	Retrieval of indicators				
	mAP	MRR	P@1	P@5	P@9
phoneme	0.912	0.984	0.993	0.981	0.919
Chinese character	0.846	0.959	0.979	0.955	0.859

The experimental results are displayed in Table 4. The above experimental results show that the model is able to achieve better performance using phonemes as pre-trained labels compared to using Chinese characters as pre-trained labels for speech recognition. The reason for this is that the features learned by the model using phonemes as pre-trained labels for speech recognition are more specific, better generalized and more applicable than those learned using words as labels. Therefore, when tested on the test set, its performance is better than that of using linguistic text as labels.

4.6. The Effect of Downsampling Modules on Speech Retrieval

This section verifies the effectiveness of the downsampling module by comparing two systems.

1. No downsampling module: the features obtained from the encoder go directly to the decoder without any change in its time step.

2. Convolutional pooling downsampling: features obtained from the encoder are downsampled by one-dimensional convolution and maximum pooling before entering the decoder.

3. Causal convolutional pooling downsampling: features obtained from the encoder are downsampled by causal convolution and maximum pooling before entering the decoder.

Table 5. Impact of the downsampling module on retrieval performance

Downsampling Module	Retrieval of indicators				
	mAP	MRR	P@1	P@5	P@9
not have	0.683	0.841	0.933	0.854	0.718
convolution pooling	0.894	0.969	0.981	0.953	0.905
causal convolution pooling	0.912	0.984	0.993	0.981	0.919

The experimental results are shown in Table 5, compared with the model without downsampling module, the use of downsampling module significantly improves the model performance, and the use of causal convolution can obtain better performance than ordinary convolution. CTC recognition model in the recognition task will be recognized and classified at each time step, and its classification results will be repeated in a large number of neighboring time steps, and the features obtained by the pre-training module will be in such a situation, too long and containing redundant information will affect the training of subsequent modules. The features obtained through its pre-training module will also be in this situation, and features that are too long and contain redundant information will affect the training of the subsequent modules, so sequence integration and feature selection through the downsampling module can improve the performance of the model.

4.7. Effect of Different Hash Code Lengths on Retrieval Performance

This section tests the effect of using different hash code lengths on the model performance, specifically examining the lengths [512,640,768,896,1024]. As analyzed in Figure 3.6, the model performance was significantly improved when increasing the hash code length from 512 to 768 bits. However, by continuing to increase the length to 1024, the retrieval accuracy starts to fluctuate.

Table 6. Effect of hash code length on retrieval performance

hash code length	Retrieval of indicators				
	mAP	MRR	P@1	P@5	P@9
512	0.873	0.967	0.984	0.966	0.884
640	0.886	0.952	0.987	0.971	0.895
768	0.912	0.984	0.993	0.981	0.919
896	0.913	0.983	0.992	0.983	0.919
1024	0.911	0.988	0.988	0.979	0.918

5. Conclusion

In this study, we introduce a novel end-to-end hashing algorithm designed specifically for speech retrieval. Unusually, the algorithm bypasses the traditional transcription phase. It directly encodes variable-length speech content at the acoustic level into a uniform-length hash code, thus eliminating the reliance on ASR accuracy. Our comprehensive evaluation highly highlights the crucial role of the pre-training, downsampling and sequence matching encoder modules in the algorithm. Moreover, the adaptability and excellent performance of the algorithm under various noise conditions make it well suited for retrieval tasks. The results of the study show that the mAP metrics of the proposed method are well adapted, emphasizing its capability in speech

retrieval applications.

In the future, in addition to using text-labeled datasets, we will apply our algorithm to unlabeled datasets. We will use an unsupervised approach to facilitate speech retrieval in situations where languages are poorly known and resources are limited.

References

[1] M. B. Ak¸cay and K. O˘ guz, "Speech emotion recognition: Emotional models, databases, features, preprocessing methods, supporting modal ities, and classifiers," Speech Communication, vol. 116, pp. 56–76, 2020. W.-K. Chen, Linear Networks and Systems (Book style). Belmont, CA: Wadsworth, 1993, pp. 123–135.

[2] L.-s. Lee, J. Glass, H.-y. Lee, and C.-a. Chan, "Spoken content retrieval—beyond cascading speech recognition with text retrieval," IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 23, no. 9, pp. 1389–1420, 2015 B. Smith, "An approach to graphs of linear forms (Unpublished work style)," unpublished.

[3] Y.-b. Huang, Y. Wang, H. Li, Y. Zhang, and Q.-y. Zhang, "Encrypted speech retrieval based on long sequence biohashing," Multimedia Tools and Applications, vol. 81, no. 9, pp. 13065–13085, 2022. J. Wang, "Fundamentals of erbium-doped fiber amplifiers arrays (Periodical style—Submitted for publication)," IEEE J. Quantum Electron., submitted for publication.

[4] W. Khan and K. Kuru, "An intelligent system for spoken term detection that uses belief combination," IEEE Intelligent Systems, vol. 32, no. 1, p. 70–79, Feb 2017. [Online]. Available: Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, "Electron spectroscopy studies on magneto-optical media and plastic substrate interfaces (Translation Journals style)," IEEE Transl. J. Magn. Jpn., vol. 2, Aug. 1987, pp. 740–741 [Dig. 9th Annu. Conf. Magnetics Japan, 1982, p. 301].

[5] T. K. Chia, K. C. Sim, H. Li, and H. T. Ng, "A lattice-based approach to query-by-example spoken document retrieval," in Proceedings of the J. U. Duncombe, "Infrared navigation—Part I: An assessment of feasibility (Periodical style)," IEEE Trans. Electron Devices, vol. ED-11, pp. 34–39, Jan. 1959. 31st annual international ACM SIGIR conference on Research and de velopment in information retrieval, 2008, pp. 363–370.

[6] C. Parada, A. Sethy, and B. Ramabhadran, "Query-by-example spoken term detection for oov terms," in 2009 IEEE Workshop on Automatic Speech Recognition & Understanding. IEEE, 2009, pp. 404–409. R. W. Lucky, "Automatic equalization for digital communication," Bell Syst. Tech. J., vol. 44, no. 4, pp. 547–588, Apr. 1965.

[7] Y. Moriya and G. J. Jones, "Improving noise robustness for spoken content retrieval using semi-supervised asr and n-best transcripts for bert-based ranking models," in 2022 IEEE Spoken Language Technology Workshop (SLT). IEEE, 2023, pp. 398–405. G. R. Faulhaber, "Design of service systems with priority reservation," in Conf. Rec. 1995 IEEE Int. Conf. Communications, pp. 3–8.

[8] W. Shen, C. M. White, and T. J. Hazen, "A comparison of query by-example methods for spoken term detection," MASSACHUSETTS INST OF TECH LEXINGTON LINCOLN LAB, Tech. Rep., 2009. G. W. Juette and L. E. Zeffanella, "Radio noise currents n short sections on bundle conductors (Presented Conference Paper style)," presented at the IEEE Summer power Meeting, Dallas, TX, Jun. 22–27, 1990, Paper 90 SM 690-0 PWRS.

[9] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition," IEEE transactions on acoustics, speech, and signal processing, vol. 26, no. 1, pp. 43–49, 1978. J. Williams, "Narrow-band analyzer (Thesis or Dissertation style)," Ph.D. dissertation, Dept. Elect. Eng., Harvard Univ., Cambridge, MA, 1993.

[10] X. Anguera and M. Ferrarons, "Memory efficient subsequence dtw for query-by-example spoken term detection," in 2013 IEEE International Conference on Multimedia and Expo (ICME). IEEE, 2013, pp. 1–6. J. P. Wilkinson, "Nonlinear resonant circuit devices (Patent style)," U.S. Patent 3 624 12, July 16, 1990.

[11] H. Kamper, W. Wang, and K. Livescu, "Deep convolutional acous tic word embeddings using word-pair side information," in 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2016, pp. 4950–4954 Letter Symbols for Quantities, ANSI Standard Y10.5-1968.

[12] C. Jacobs, Y. Matusevych, and H. Kamper, "Acoustic word embeddings for zero-resource languages using self-supervised contrastive learning and multilingual adaptation," in 2021 IEEE Spoken Language Technology Workshop (SLT). IEEE, 2021, pp. 919–926. E. E. Reber, R. L. Michell, and C. J. Carter, "Oxygen absorption in the Earth's atmosphere," Aerospace Corp., Los Angeles, CA, Tech. Rep. TR-0200 (420-46)-3, Nov. 1988.

[13] H. Kamper, Y. Matusevych, and S. Goldwater, "Improved acoustic word embeddings for zero-resource languages using multilingual transfer," IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 29, pp. 1107–1118, 2021.

[14] Q.-y. Zhang, X.-j. Zhao, Q.-w. Zhang, and Y.-z. Li, "Content-based encrypted speech retrieval scheme with deep hashing," Multimedia Tools and Applications, p. 10221–10242, Mar 2022.

[15] Y. Yuan, L. Xie, C.-C. Leung, H. Chen, and B. Ma, "Fast query-by example speech search using attention-based deep binary embeddings," IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 28, pp. 1988–2000, 2020.

[16] S.-W. Fan-Jiang, T.-H. Lo, and B. Chen, "Spoken document retrieval leveraging bert-based modeling and query reformulation," in ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Sig nal Processing (ICASSP). IEEE, 2020, pp. 8144–8148.

[17] H. Muaidi, A. Al-Ahmad, T. Khdoor, S. Alqrainy, and M. Alkof fash, "Arabic audio news retrieval system using dependent speaker mode, mel frequency cepstral coefficient and dynamic time warp ing techniques," Research Journal of Applied Sciences, Engineer ing and Technology, p. 5082–5097, Oct 2016. [Online]. Available: http://dx.doi.org/ 10.19026/ rjaset. 7.903.

[18] F. Shen, C. Du, and K. Yu, "Acoustic word embeddings for end-to end speech synthesis," Applied Sciences, p. 9010, Sep 2021 [Online]. Available:http:// dx.doi.org/ 10.3390/ app 1119901.

[19] A. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "Wavenet: A gener ative model for raw audio," en-USSSW,SSW, Sep 2016.

[20] C. Lea, M. D. Flynn, R. Vidal, A. Reiter, and G. D. Hager, "Tempo ral convolutional networks for action segmentation and detection," in proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 156–165.

[21] H. Wang, F. Gao, Y. Zhao, L. Yang, J. Yue, and H. Ma, "Multitask learning with local attention for tibetan speech recognition," Complex ity, vol. 2020, pp. 1–10, 2020.

[22] B.-H. Sung and S.-C. Wei, "Becmer: A fusion model using bert and cnn for music emotion recognition," in 2021 IEEE 22nd International Con ference on Information Reuse and Integration for Data Science (IRI). IEEE, 2021, pp. 437–444.

[23] J. Mingyu, Z. Jiawei, and W. Ning, "Afr-bert: Attention-based mech anism feature relevance fusion multimodal sentiment analysis model," Plos one, vol. 17, no. 9, p. e0273936, 2022.

[24] D. Wang and X. Zhang, "Thchs-30: A free chinese speech corpus," arXiv preprint arXiv:1512.01882, 2015.