A Method Improves Speech Recognition with Contrastive Learning in Low-Resource Languages

Lixu Sun 1,2, Nurmemet Yolwas 1,2,* and Lina Jiang 1,2

Abstract: Building an effective automatic speech recognition system typically requires a large amount of high-quality labeled data; however, this can be challenging for low-resource languages. Currently, self-supervised contrastive learning has shown promising results in low-resource automatic speech recognition, but there is no discussion on the quality of negative sample sets in speech contrastive learning. In this paper, we propose the false negatives impact elimination (FNIE) method to filter false negative samples and improve the quality of the negative sample set in speech. FNIE compares the support vector with the negative sample vector set and optimizes the corresponding loss function, allowing the model to learn better speech representations and achieve superior results in low-resource speech recognition. Experiments demonstrate that FNIE effectively filters negative samples, enhances the quality of the negative sample set, and improves the accuracy of speech recognition. The quality of the negative sample set significantly affects the model’s learning ability, and using too many negative samples can deteriorate it. In a low-resource setting, our FNIE method achieved a relative improvement of 2.98% in WER on the English dataset, 14.3% in WER on the Uyghur dataset, and 4.04% in CER on the Mandarin dataset compared to the baseline model.

Keywords: contrastive learning; negative samples; wav2vec 2.0; low-resource

1. Introduction

Automatic speech recognition is designed to realize the transformation from speech sequences to text sequences. In recent years, compared with the architectures of traditional automatic speech recognition [1], the end-to-end frameworks have shown better recognition effects in the field of speech recognition [2–5]. Unlike traditional automatic speech recognition models, which consist of acoustics, pronunciation, and language modeling, the end-to-end model has a simpler architecture and a more streamlined training process [6]. However, the current end-to-end model requires a large amount of high-quality labeled data to train the model in order to achieve a good recognition result. This poses a very significant challenge for low-resource languages that lack sufficient labeled data to train end-to-end models [7]. Therefore, it is worthwhile to explore methods for utilizing limited labeled speech data to improve the performance of low-resource automatic speech recognition systems.

The current mainstream approach to solve this problem is to use self-supervised learning methods. Self-supervised learning focuses on extracting supervised information from large-scale unlabeled data by using auxiliary tasks, and training the network with this constructed supervised information. This enables the network to learn representations that are useful for downstream tasks [8,9]. Among them, contrastive learning plays a great role in the field of self-supervised learning, which constructs representations by learning to encode the similarity or dissimilarity of two things [10–12]. It looks at which pairs of data points are “similar” and “different” in order to learn higher-level features, and
has a wide range of applications in many fields, such as computer vision [13], emotion recognition [14], natural language processing [15], and domain adaptation [16]. The quality of the representations learned by contrastive learning is largely influenced by the quality of negative samples. In the field of computer vision, two properties of good computer vision representations are alignment and uniform distribution [17], and there are also many ways to select positive samples [18], filter negative samples [19], increase the number of negative samples [20], and generate harder negative samples [21]. A similar approach is used in natural language processing to find high-quality sets of negative samples [21,22].

Many models using contrastive learning methods have also been proposed in the field of automatic speech recognition. Contrastive Predictive Coding (CPC) [23] uses an autoregressive model and noise-contrastive estimation [24] to discard the lower-level information and noise at the lower levels and extract the higher-dimensional speech representations to predict future information. wav2vec [25] proposes a noise-contrast learning binary classification task using the ideas proposed by CPC, thus allowing wav2vec to train on a large amount of unlabeled data and achieve better feature extraction results in speech recognition tasks. Based on wav2vec, vq-wav2vec [26] realizes the conversion process of feature space from infinite continuous to finite discrete through the proposed quantization module, replaces traditional acoustic features with speech representation generated by combination with BERT, and then trains the speech recognition model in a supervised manner, which improves the latest performance of The Wall Street Journal and TIMIT benchmarks. wav2vec 2.0 [27] uses comparative learning to propose a new self-supervised architecture, combining the Gumbel softmax quantization module [28] and BERT in vq-wav2vec into a model, which can be fine-tuned directly in speech downstream tasks with only one step of pre-training, and shows the feasibility of speech recognition with low resources. JUST [29] builds on wav2vec 2.0 with self-supervised use of contrastive loss and MLM loss and supervised use of RNN-T loss [30] for joint training to achieve higher accuracy in multilingual low-resource situations. wav2vec-S [31] proposes use of the semi-supervised pre-training method of wav2vec 2.0 to build a better low-resource speech recognition pre-training model.

Many relevant contrastive learning models have been proposed in low-resource speech recognition, but the selection of negative samples for speech remains unmentioned. Changes in speech are continuous and minor within a small range. Thus, the approach of defining negative samples as samples from different parts of the same speech tends to ignore the fact that they may also have the same pronunciation as the positive samples. This subtle change is still very important for low-resource speech recognition. Therefore, we propose the false negatives impact elimination (FNIE) method to discover potential false negative samples in speech contrastive learning and optimize the corresponding loss function to eliminate the impact of false negative samples, which improves the quality of the negative sample set and thus allows the model to learn better speech representations. We evaluate the results of our method for English, Mandarin, Uyghur, and Uzbek in the low-resource case and compare the effect of the number of negative samples on the model. The experiments show that our method learns better speech representations and achieves higher accuracy in low-resource speech recognition compared to the original model for the same number of elements in the final negative sample set. The FNIE method can also further improve the accuracy of low-resource speech recognition when increasing the number of negative samples deteriorates the learning effect.

The main contributions of this paper are as follows:

1. We propose the false negatives impact elimination (FNIE) method and optimize the corresponding loss function to improve the quality of the negative sample set of speech, allowing the model to learn better speech representations and achieve better results in low-resource speech recognition;
2. We found that simply increasing the number of negative samples does not improve the model’s ability to learn speech representations;
3. By training our model on a variety of low-resource languages, we have validated its feasibility and effectiveness.

2. Related Work

The research on contrastive learning and low-resource learning is a rapidly growing field with numerous promising approaches and applications [32–35]. In this section, we provide a brief overview of the research related to speech contrastive learning, which can be divided into three parts: CPC, wav2vec, and wav2vec 2.0.

2.1. Contrastive Predictive Coding

Aaron et al. proposed a general unsupervised comparative learning framework. Its purpose is to extract useful representation information from high-dimensional sequence data, compare the context representation information extracted from the data with the representation information of samples at future time, and obtain the key representation information that best predicts the future, which is called Contrastive Predictive Coding (CPC). The model first cuts the audio sequence into suitable sequence fragments \( X = x_1, \ldots, x_T \), and the fragments are mapped into the hidden space \( Z \) using a nonlinear encoder \( g_{\text{enc}} \) with \( z_t = g_{\text{enc}}(x_t) \). The autoregressive model \( g_{\text{ar}} \) is then used to fuse the historical temporal information \( z \) to obtain the context representation \( C, c_t = g_{\text{ar}}(z_{\leq t}) \). The model architecture of Contrastive Predictive Coding using speech as input is shown in Figure 1.

![Figure 1. Overview of Contrastive Predictive Coding, audio as input.](image)

The encoder and autoregressive models are optimized using InfoNCE loss. At each time \( t \) there is given a set \( X = x_1, \ldots, x_N \) of \( N \) random samples containing one positive sample from \( p(x_{t+k}|c_t) \) and \( N - 1 \) negative samples from the proposal distribution \( p(x_{t+k}) \), with the following loss function

\[
L_N = -\mathbb{E}_X \left[ \log \frac{\exp(z_t^T h_k(c_t))}{\sum_{x_j \in X} \exp(z_j^T h_k(c_t))} \right]
\]

where \( h_k(\cdot) \) is a linear transformation for every step \( k \).

2.2. wav2vec

wav2vec is an unsupervised pre-training method for speech recognition that learns representations of raw audio. It trains on a large amount of unlabeled data to learn speech representations. The resulting representations are then put into an acoustic model instead of traditional acoustic features such as log-mel filterbank features. wav2vec consists of two main modules: the encoder network and the context network. First, give an original voice signal \( x_i \in X \), using the encoder to generate a low-frequency feature representation \( z_i \in Z \).
Then mix multiple latent representations $z_i...z_i-v$ as the context inputs produce a single contextualized tensor $c_i = g(z_i...z_i-v)$. The framework of the model is shown in Figure 2.

\[
L_k = -\sum_{i=1}^{T-k} \left( \log \sigma(z_{i+k}h_k(c_i)) + \lambda \mathbb{E}_{z \sim p_n} \left[ \log \sigma(-z^T_h(c_i)) \right] \right)
\]

where $\sigma(x) = 1/(1 + \exp(-x))$, $h_k(\cdot)$ is a step-specific affine transformation for every step $k$, $T$ is the length of the audio sequence, and $\lambda$ is the number of negative samples.

2.3. wav2vec 2.0

The basic framework of wav2vec 2.0 is illustrated in Figure 3. This framework is designed for self-supervised learning of representations from raw audio data. The original speech audio $X$ is encoded by a multilayer convolutional neural network, and then mask the generated late speech representations with the same method as masked language modeling $Z = z_1,...,z_t$. The masked latent speech representations are fed to a Transformer network to build contextualized representations $C = c_1,...,c_t$, and fed to a quantization network to generate quantized representations $Q$, to calculate contrastive loss and optimize the model.
Its corresponding contrast loss is

\[
L_m = -\log \frac{\exp(\text{sim}(c_t, q_t)/\kappa)}{\sum_{\tilde{q} \sim Q} \exp(\text{sim}(c_t, \tilde{q})/\kappa)}
\]

where \( \text{sim}(a, b) = a^T b / ||a|| ||b|| \) is used to calculate the cosine similarity of contextualized representations and quantized representations, and \( \kappa \) is a super parameter.

3. Methods

3.1. Model

The process of our method is shown in Figure 4. The original audio signal \( X \) is passed through encoder to extract features and generate latent speech representations \( Z = z_1, ..., z_t \) for \( t \) time-steps. Before \( Z \) is fed into the Transformer and Quantization modules, we randomly select starting time steps for the mask with a proportion of \( p \) across all time steps, and then mask the consecutive \( M \) time steps for the selected indices. During the masking operation, the latent speech representations \( Z \) at each time step can be considered as a starting time step, so the selected mask span may overlap. We put \( Z \) into Transformer and Quantization by masking operation to produce contextual vectors \( C = c_1, ..., c_i \) and quantized representation \( Q = q_1, ..., q_i \). The \( q_i \) of the corresponding step is used as a positive sample of the contextual vector \( c_i \), then \( K \) quantized vectors are randomly selected from \( Q \) as the negative sample \( \bar{Q} \) of \( c_i \). Put the unmasked \( Z \) into Transformer with no gradient calculation, then masking the output vectors with the same mask to produce support vectors \( S = s_1, ..., s_i \). \((S, \bar{Q})\) eliminates the false negative samples by the FNIE and generates a new set of \( K' \) negative samples \( \bar{Q}' \), which perform the calculation of the contrast loss.

We hypothesize that the main difference between negative and false negative samples lies in the fact that false negative samples have a higher degree of similarity to positive samples, which can lead the model to learn in the wrong direction. To address this issue, we propose the FNIE method, which filters out false negative samples that exhibit significant similarities with positive samples.

![Figure 4. Illustration of the optimized model framework. Support Vectors and Negatives calculate the similarity and take the top-k similarity as false negative samples, which are deleted or absorbed as positive samples.](image-url)
3.2. FNIE

Due to the nature of speech, the selection of distractors may have a risk of approaching context representation and being similar to quantized representation. We propose the false negatives impact elimination method and modify the corresponding loss function to alleviate this problem. The FNIE method includes a strategy for detecting false negative samples, and a method for using false negatives to improve contrastive learning. We propose the method as shown in Figure 5.

3.2.1. Finding False Negative Samples

Although the pronunciation of the same phonemic units may be affected by different factors, the change can be minimized within a certain time window due to the characteristics of speech. Based on this idea, we propose a strategy for identifying potential false negative samples as follows:

1. For each time step $t$, the original audio feature $z_t$ is transformed by a Transformer function $g: Z \rightarrow C^a$ to produce the support vector $c^a_t$.
2. For each negative sample $\bar{q}_i, k \in \bar{Q}$ and each support vector $s_i$, calculate the similarity score $\text{score}_{i,k} = \text{sim}(s_i, \bar{q}_i, k)$.
3. For each negative sample and the corresponding support vector, calculate the similarity and sort the scores in descending order: $\text{score}_{\text{dec}, i,k} = \text{sort}_{s_i \in S}(\text{score}_{i,k})$.
4. Define a set of potential false negative samples $F_{i,N}$, where the negative samples are the $N$ most similar to the support vector at that step, $F_{i,N} = \{\bar{q}_{i,k} | \text{top}(\text{score}_{\text{dec}, i,k}, N)\}$, and $\bar{Q}' = \bar{Q}_i - F_{i,N}$.

where $\text{sim}(a, b) = a^Tb/|a||b|$, and the Transformer must stop gradient calculation in this step. As shown in Figure 6, we provide a sample of finding parts for the FNIE.

3.2.2. Delete

With FNIE, we find the false negative samples that exist in speech contrastive learning. To cooperate with FNIE, we propose the following loss function accordingly.

$$L_{del} = -\log \frac{\exp(\text{sim}(c_i, q_i)/\tau)}{\sum_{q \in Q} \exp(\text{sim}(c_i, q)/\tau)}$$

(4)

We believe that the delete method is very stable when the final number of negative samples is the same, and it can effectively mitigate the interference from the same utterance with the
same pronunciation from different parts. By this method, the model can exclude the effect of false negative samples, which enables the model to learn better phonetic representations and achieve better results in the subsequent downstream tasks.

Figure 6. A sample of finding parts for the FNIE. The Transformer Encoder generates support vectors, and quantization generates negatives. The set of negative samples used for the final loss value calculation is generated by FNIE.

3.2.3. Assimilate

Eliminating false negative samples can indeed mitigate the negative impact of contrast, but this process still has shortcomings in ignoring information similar to false negative samples and positive samples. False negative samples may come from the same phonetic unit corresponding to different words, so we believe that false negative samples that have been identified can be treated as positive samples. However, the number of identical sounds in a sentence is always limited, so they must have different weights for positive sample processing even if the samples are in the same set $F_i,N$. We propose the following loss functions:

$$L_{ass} = -\left\{ \gamma \log \frac{\exp\left( \frac{\text{sim} (c_i, q_i)}{\tau} \right)}{\sum_{q \in Q^T} \exp\left( \frac{\text{sim} (c_i, q_i)}{\tau} \right)} + a \log \frac{\exp\left( \frac{\text{sim} (c_i, q_i)}{\tau} \right)}{\sum_{q \in F_i,1} \exp\left( \frac{\text{sim} (c_i, q_i)}{\tau} \right)} + \epsilon \log \frac{\exp\left( \frac{\text{sim} (c_i, q_i)}{\tau} \right)}{\sum_{q \in F_i,2} \exp\left( \frac{\text{sim} (c_i, q_i)}{\tau} \right)} \right\}$$

(5)
where if $N > 1$, then $\gamma + \alpha + \varepsilon = 1$ and if $N = 1$, then $\gamma + \alpha = 1, \varepsilon = 0$. Compared with the simple deletion of false negative samples, the assimilation method has higher requirements for the quality of selected false negative samples, and the assimilation of excessive false negative samples will seriously degrade the learning quality of the pre-trained model. The similarity is different in assimilation, and we believe its impact on loss function is different. In addition to the positive samples, the most similar false negative samples should occupy a higher proportion, and the false negative samples in the remaining set of false negative samples should occupy a relatively small proportion.

4. Experimental Methods

4.1. Corpus

In this paper, we used speech datasets of four languages to complete our experiments, namely, Mandarin (MA), English (EN), Uyghur (Uy), and Uzbek (Uz). The dataset details are shown in the following Table 1.

<table>
<thead>
<tr>
<th>Datasets</th>
<th>Language</th>
<th>Durations</th>
<th>Format</th>
<th>Sampling Rate</th>
<th>Speakers</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>LibriSpeech-train-clean-100 h</td>
<td>English</td>
<td>100 h</td>
<td>FLAC</td>
<td>16 kHZ</td>
<td>251</td>
<td>subword</td>
</tr>
<tr>
<td>AISHELL-1</td>
<td>Mandarin</td>
<td>150 h</td>
<td>WAV</td>
<td>16 kHZ</td>
<td>340</td>
<td>character</td>
</tr>
<tr>
<td>THCHS-30</td>
<td>Mandarin</td>
<td>33.5 h</td>
<td>WAV</td>
<td>16 kHZ</td>
<td>40</td>
<td>character</td>
</tr>
<tr>
<td>Common Voice Corpus 11.0-Uyghur</td>
<td>Uyghur</td>
<td>113 h</td>
<td>MP3</td>
<td>32 kHZ</td>
<td>747</td>
<td>subword</td>
</tr>
<tr>
<td>Common Voice Corpus 11.0-Uzbek</td>
<td>Uzbek</td>
<td>258 h</td>
<td>MP3</td>
<td>32 kHZ</td>
<td>2025</td>
<td>subword</td>
</tr>
</tbody>
</table>

The English speech dataset used in this study is the Librispeech dataset [36], which is an audiobook dataset containing both text and speech data. It consists of approximately 1000 h of 16 kHz speech data, which are segmented, trimmed, aligned, and organized into audio files of approximately 10 s duration along with their corresponding text annotations. For pre-training, we use the train-clean-100 subset of the dataset, which includes around 100 h of speech data and corresponding text annotations, out of which 20 h are randomly selected for fine-tuning. The test-clean subset, consisting of 5.4 h of speech data and corresponding text annotations, is used as the test dataset.

The Mandarin speech datasets used were AISHELL-1 [37] and THCHS-30 [38]. The AISHELL-1-train subset was used as the model pre-trained unlabeled speech data, which contains approximately 150 h of 16 kHz speech data, 340 speakers, and 120,098 utterances. The THCHS-30 train was used as the fine-tuning dataset, and THCHS-30-test was used as the test dataset.

For Uyghur, we used the validated part of Common Voice Corpus 11.0-Uyghur [39], a subset containing about 107 h of speech and corresponding text data, of which 20 h were randomly selected as a fine-tuning dataset and 5 h as a test dataset.

For Uzbek, we used the validated part of Common Voice Corpus 11.0-Uzbek [39] was used for Uzbek. This subset contains about 97 h of speech and corresponding text data, of which 20 h were randomly selected as a fine-tuning dataset and 5 h as a test dataset.

4.2. Implementation Details

All experiments were conducted using fairseq [40] and were conducted on 2 NVIDIA RTX A5000 Graphics Cards. For the pre-training stage, we used the open-source wav2vec 2.0 base model. The CNN layer in our model comprises of 7 hidden layers, where each layer includes a temporal convolution, layer normalization, and a GELU activation function. The temporal convolution of each block has 512 channels, the width of the convolution kernel is $(10, 3, 3, 3, 2, 2, 2)$, the stride size is $(5, 2, 2, 2, 2)$, the stride length is about 20 ms, and the receptive field is 25 ms. The self-attention layer consists of a 12-layer, 768-dimensional Transformer layer with 8 self-attention heads. The encoder layer dropout is 0.5, and dropout input and dropout features are 0.1. For the mask operation, the value of
\( p \) is 0.065, and \( M \) is 10. The quantization module has two codebooks with 320 entries in each, and each entry has a dimension of 128. For English, Uyghur, and Uzbek, we used gradient accumulation to implement a batch size of 11.2 m samples, and for Mandarin, we used gradient accumulation to implement a batch size of 22.4 m samples, with the peak learning rate 0.0001 and warm-up steps 32,000. The pre-training is 100 epochs, batch size is 8, and the other parameters were basically the same as the base model configuration in [27]. For the loss function (3), we set \( a = 0.1 \), and \( \epsilon = 0.05 \). The network structure of the model is shown in Figure 7.

In the fine-tuning stage, Mandarin fine-tuned for 100 epochs, Uyghur fine-tuned for 200 epochs, Uzbek fine-tuned for 400 epochs, and English fine-tuned for 30 k steps. Using gradient accumulation to implement a batch size of 7.68 m samples for English, Mandarin, and Uyghur, using gradient accumulation to implement a batch size of 5.76 m samples for Uzbek, with the peak learning rate 0.00005, 8 batch size, and warm-up steps 3000, and the other parameters were basically the same as the base model configuration in the fine-tune experiments of [27]. The same language uses the same language model, where EN language model is 4-g from LibriSpeech-train-clean-960h text made by kenLM [41], MA language model is 4-g from The People’s Daily text made by kenLM, and UG and UZ are 3-g from respective text made by kenLM.

5. Experimental Results

In this paper, we use wav2vec 2.0 framework as a baseline system for pre-training and self-supervised comparative learning pre-training for different languages in the proposed method. By training different numbers of \( N, K, K' \), it can be demonstrated that our proposed FNIE method can learn speech representations more effectively and is equally applied in multiple languages.
5.1. Results of Deleting False Negative Samples

Our experimental results are shown in Table 2. We found that deleting a small number of false negative samples had a positive effect on the model’s optimization and improved the accuracy of the model in low-resource speech recognition. However, deleting too many false negative samples had a negative effect on the model’s optimization. In the English dataset, deleting two false negative samples resulted in a relative improvement of 2.69% compared to the baseline model. In Mandarin, deleting four false negative samples resulted in a 3.94% improvement relative to the original model. However, if too many false negative samples are deleted, it can lead to the deletion of difficult negative samples, which can negatively affect the model’s ability to learn speech representation by reducing its distance from the embedding space for negative samples. We believe that this occurs because negative samples with similar but slightly different pronunciations from $c_i$ are removed, reducing the quality of the negative sample set and affecting the model’s learning of difficult negative samples. This, in turn, leads to a slight reduction in the model’s ability to learn speech representations. If the speech data contain more but shorter segments, then the number of false negative samples that need to be deleted should be increased accordingly.

Table 2. Comparison of deleting different numbers of false negative samples.

<table>
<thead>
<tr>
<th>Method</th>
<th>$K$</th>
<th>$K'$</th>
<th>$N$</th>
<th>Mandarin CER%</th>
</tr>
</thead>
<tbody>
<tr>
<td>wav2vec 2.0</td>
<td>100</td>
<td>100</td>
<td>0</td>
<td>27.853</td>
</tr>
<tr>
<td>FNIE</td>
<td>101</td>
<td>100</td>
<td>1</td>
<td>27.791</td>
</tr>
<tr>
<td></td>
<td>102</td>
<td>100</td>
<td>2</td>
<td>27.335</td>
</tr>
<tr>
<td></td>
<td>104</td>
<td>100</td>
<td>4</td>
<td>26.755</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method</th>
<th>$K$</th>
<th>$K'$</th>
<th>$N$</th>
<th>English CER%</th>
<th>WER%</th>
</tr>
</thead>
<tbody>
<tr>
<td>wav2vec 2.0</td>
<td>100</td>
<td>100</td>
<td>0</td>
<td>11.753</td>
<td>23.996</td>
</tr>
<tr>
<td>FNIE</td>
<td>101</td>
<td>100</td>
<td>1</td>
<td>12.115</td>
<td>24.062</td>
</tr>
<tr>
<td></td>
<td>102</td>
<td>100</td>
<td>2</td>
<td>11.805</td>
<td>23.353</td>
</tr>
<tr>
<td></td>
<td>104</td>
<td>100</td>
<td>4</td>
<td>12.18</td>
<td>24.254</td>
</tr>
</tbody>
</table>

5.2. Results of Assimilating False Negative Samples

Compared to the deletion strategy, the assimilation strategy is more sensitive to the quality of the false negative samples. As shown in Table 3, in the English dataset, when assimilating one false negative sample, there is a relative improvement of 2.98% over the original model, and in the Mandarin dataset, a relative improvement of 4.04% is still obtained, with the final number of negative samples remaining the same. However, after assimilation of two false negative samples, both the English and Mandarin models have a negative effect on the model. That is, the assimilation strategy is only suitable for more reliable false negative samples (i.e., those with very high similarity scores), and excessive false negative samples will interfere with the model, making it difficult for the model to capture the characteristics of positive samples. This is consistent with our original assumption that the negative sample with the highest similarity will improve the model’s learning ability and have a larger weight, while other samples assimilated as positive samples will have a smaller weight. Compared to the method of deleting false negative samples, the method of assimilating false negative samples can achieve higher accuracy in longer speech segments. However, this method requires higher quality for the false negative samples.
Table 3. Comparison of assimilating different numbers of false negative samples.

<table>
<thead>
<tr>
<th>Method</th>
<th>K</th>
<th>K’</th>
<th>N</th>
<th>Mandarin CER%</th>
</tr>
</thead>
<tbody>
<tr>
<td>wav2vec 2.0</td>
<td>100</td>
<td>100</td>
<td>0</td>
<td>27.853</td>
</tr>
<tr>
<td>FNIE</td>
<td>101</td>
<td>100</td>
<td>1</td>
<td>26.725</td>
</tr>
<tr>
<td></td>
<td>102</td>
<td>100</td>
<td>2</td>
<td>27.9</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method</th>
<th>K</th>
<th>K’</th>
<th>N</th>
<th>English</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>CER%</td>
</tr>
<tr>
<td>wav2vec 2.0</td>
<td>100</td>
<td>100</td>
<td>0</td>
<td>11.753</td>
</tr>
<tr>
<td>FNIE</td>
<td>101</td>
<td>100</td>
<td>1</td>
<td>11.694</td>
</tr>
<tr>
<td></td>
<td>102</td>
<td>100</td>
<td>2</td>
<td>11.862</td>
</tr>
</tbody>
</table>

5.3. Discussion on the Number of Negative Samples

We compared the effects of using different numbers of negative samples on our model and present the results in Table 4. From the table, it is evident that simply increasing the number of negative samples does not necessarily improve the model’s performance. In fact, it may worsen the model’s ability to learn representation and reduce the accuracy of speech recognition. Therefore, we suggest that improving the quality of the set of negative samples is the right approach to enhance the model’s representation learning ability and improve the speech recognition effect. Increasing the number of negative samples alone does not necessarily improve the model’s representation learning ability. This also shows that it is possible to improve low-resource language speech recognition effects using our method.

Table 4. Comparison of different numbers of negative samples.

<table>
<thead>
<tr>
<th>Method</th>
<th>loss</th>
<th>K</th>
<th>K’</th>
<th>N</th>
<th>English</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FNIE</td>
<td>ass</td>
<td>101</td>
<td>100</td>
<td>1</td>
<td>11.694</td>
</tr>
<tr>
<td></td>
<td>del</td>
<td>102</td>
<td>100</td>
<td>2</td>
<td>11.805</td>
</tr>
<tr>
<td>wav2vec 2.0</td>
<td>-</td>
<td>100</td>
<td>100</td>
<td>0</td>
<td>11.753</td>
</tr>
<tr>
<td></td>
<td>-</td>
<td>101</td>
<td>101</td>
<td>0</td>
<td>11.294</td>
</tr>
<tr>
<td></td>
<td>-</td>
<td>102</td>
<td>102</td>
<td>0</td>
<td>11.328</td>
</tr>
<tr>
<td></td>
<td>-</td>
<td>104</td>
<td>104</td>
<td>0</td>
<td>11.517</td>
</tr>
</tbody>
</table>

5.4. Results of Different Languages

To validate the effectiveness and reliability of the proposed method, we conducted experiments on Uyghur and Uzbek languages, and the results are shown in Table 5. On Uyghur and Uzbek languages, our proposed method achieved relatively better experimental results. Because of the short duration of the audio data for the two languages mentioned above, for the selection of negative samples, the original model is more likely to select negative samples with similar pronunciation for ci. This results in a relatively low quality of the negative sample set, but using the FNIE method can significantly improve the quality of the negative sample set. We believe that the assimilation of false negative samples is more effective in improving the model’s ability to learn speech representations for longer speech clips, and the deletion of false negative samples is more effective for shorter speech clips. Additionally, this experiment demonstrates that our proposed method is widely applicable and has good effectiveness and reliability, and also verifies the feasibility of our proposed method.
### Table 5. Comparison of different languages.

<table>
<thead>
<tr>
<th>Method</th>
<th>loss</th>
<th>K</th>
<th>K'</th>
<th>N</th>
<th>WER%</th>
</tr>
</thead>
<tbody>
<tr>
<td>wav2vec 2.0</td>
<td>-</td>
<td>100</td>
<td>100</td>
<td>0</td>
<td>5.76</td>
</tr>
<tr>
<td>FNIE</td>
<td>ass</td>
<td>101</td>
<td>100</td>
<td>1</td>
<td>5.423</td>
</tr>
<tr>
<td></td>
<td>del</td>
<td>102</td>
<td>100</td>
<td>2</td>
<td>4.936</td>
</tr>
<tr>
<td>wav2vec 2.0</td>
<td>-</td>
<td>100</td>
<td>100</td>
<td>0</td>
<td>16.789</td>
</tr>
<tr>
<td>FNIE</td>
<td>ass</td>
<td>101</td>
<td>100</td>
<td>1</td>
<td>12.019</td>
</tr>
<tr>
<td></td>
<td>del</td>
<td>102</td>
<td>100</td>
<td>2</td>
<td>10.902</td>
</tr>
</tbody>
</table>

6. Conclusions
In this work, we propose a method to improve the quality of the negative sample set of speech and optimize the corresponding loss function to boost the model’s ability to learn speech representations. Our proposed method leads to improved speech recognition accuracy in low-resource scenarios. Additionally, we demonstrate that simply increasing the number of negative samples does not necessarily enhance the model’s performance. To improve the model’s representation learning ability and speech recognition accuracy, it is crucial to improve the quality of the negative sample set. Our proposed method, FNIE, is feasible for multiple languages.

Our work investigates the possibility of enhancing low-resource speech recognition by improving the quality of negative sample sets. However, our work also has certain limitations, such as the potential risk of removing true negative samples while eliminating false negative samples. In future work, we will continue to explore ways to have better quality in the set of negative samples of speech in contrastive learning and aim to improve the accuracy of automatic speech recognition in low-resource languages.

Author Contributions: Conceptualization, N.Y.; Software, L.S.; Validation, L.S.; Formal analysis, L.S.; Investigation, L.J. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Natural Science Foundation of China—Research on Key Technologies of Speech Recognition of Chinese and Western Asian Languages under Resource Constraints (Grant No. 62066043), and the National Language Commission key Project—Research on Speech Keyword Search Technology of Chinese and Western Asian Languages (Grant No. ZDI135-133).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest

References
Appl. Sci. 2023, 13, 4836


\textbf{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.