Chinese Journal of Electronics
Vol.32, No.3, May 2023

A Novel Re-weighted CTC Loss for Data Imbalance in Speech Keyword Spotting

LAN Xiaotian, HE Qianhua, YAN Haikang, and LI Yanxiong

(*School of Electronic and Information Engineering, South China University of Technology, Guangzhou 510641, China*)

Abstract — Speech keyword spotting system is a critical component of human-computer interfaces. And connectionist temporal classifier (CTC) has been proven to be an effective tool for that task. However, the standard training process of speech keyword spotting faces a data imbalance issue where positive samples are usually far less than negative samples. Numerous easy-training negative examples overwhelm the training, resulting in a degenerated model. To deal with it, this paper tries to reshape the standard CTC loss and proposes a novel re-weighted CTC loss. It evaluates the sample importance by its number of detection errors during training and automatically down-weights the contribution of easy examples, the majorities of which are negatives, making the training focus on samples deserving more training. The proposed method can alleviate the imbalance naturally and make use of all available data efficiently. Evaluation on several sets of keywords selected from AISHELL-1 and AISHELL-2 achieves 16%–38% relative reductions in false rejection rates over standard CTC loss at 0.5 false alarms per keyword per hour in experiments.

Key words — Speech keyword spotting, Connectionist temporal classifier, Data imbalance, Sample importance re-weighting.

I. Introduction

Speech keyword spotting (KWS) which is also known as spoken term detection (STD) [1], refers to the task of continuously detecting keywords in an unconstrained audio stream or a pre-recorded audio archive [2]. Keyword spotter can quickly detect useful information embedded in natural conversational speech and has been widely applied in voice controlling [3], audio retrieval [4], audio monitoring [5], and so on.

Popular KWS architectures can be classified into two categories: keyword/filler posterior modeling followed by a search algorithm and end-to-end (E2E) based architectures. In the keyword/filler modeling style, each word or subword of the keyword is modeled by a hidden Markov model (HMM) and another phone-loop graph is usually used as a filler model to absorb non-keyword speech segments [6], [7]. In recent years, benefiting from the success of deep learning, neural networks-based methods [8]–[11] greatly improve the performance of KWS. And E2E based methods, such as the connectionist temporal classifier (CTC) [12] based model [13]–[15], bring further improvement to KWS and have attracted more and more interest since there are no requirements of frame-level labels and predefined alignments. CTC addresses variable-length input and output sequence and allows the network to predict label at any point of the input sequence. And CTC shows good performance in KWS as can be seen in [13]–[15]. Hence, this study uses CTC for the KWS system.

The problem of data imbalance is common in KWS training [16]–[18], where a large amount of diverse negative training samples that may have pronunciation similar to the keyword are indeed required to prevent false alarms. Simultaneously, it is easy to collect abundant negative training data, while it is expensive to collect positive keyword data. As a result, the imbalance leads to an inefficient training process as most easy examples are actually well-learned during the initial few epochs. It should be pointed out that in this study, "easy/hard examples" represent the samples that are easy/hard to train and basically do not/easily produce detection errors (including false alarms and false rejections) during training. According to [18], this data imbalance can actually be summarized as the imbalance in difficulty or importance of samples. Specifically, during decoding, false alarms are mainly caused by a small number of hard negative examples having pronunci-

Manuscript Received June 2, 2021; Accepted Feb. 3, 2022. This work was supported by the National Natural Science Foundation of China (61571192) and Guangdong Basic and Applied Basic Research Foundation (2021A1515011454).
 DOI:10.23919/cje.2021.00.198

ation similar to the keyword; however, the majority of easy negative examples overwhelm the loss and dominate the gradient backpropagation, which results in a degenerated model. To alleviate the data imbalance, Hou *et al.* [17] introduced a regional hard-example mining algorithm to select representative negative training samples to achieve a relative balance in quantity. It actually discards a lot of samples during training. Alternatively, cost-sensitive learning [19] is an effective solution that allows the use of all samples to solve the imbalance. It can be divided into class balanced re-weighting [20] and sample importance re-weighting. A typical instance of the latter is focal loss, which was first proposed to address the data imbalance of dense object detection [21] and later applied to KWS in [16]. Focal loss can automatically down-weight the contribution of easy examples during training and rapidly focus on hard examples. Zhang *et al.* [18] also used sample importance re-weighting to handle the data imbalance of KWS. Unlike [16], they re-weighted sample loss considering local interval sample difficulty instead of frame difficulty.

All the aforementioned works use loss values to evaluate the difficulty or importance of samples (loss values-guided methods). During training, in these methods, the larger the loss a sample produces, the more important it is regarded as. However, it is evident that large loss values do not always mean false detection, and small values may not represent correctness [22]. This phenomenon is more common in the CTC-based KWS system. The training criterion of CTC is to maximize the probability of the target sequence, i.e., a maximum likelihood learning process [23]. The log-likelihood reflects the probability of making the whole transcription completely correct; what it ignores are the probabilities of incorrectly transcribing keywords, resulting in all incorrect transcriptions being considered equally bad. In other words, the model is not straightly trained with the final performance metric, which is typically false alarm rate (FAR) and false rejection rate (FRR), where the FAR refers to the number of false alarms (FAs) per keyword per hour (fa/kw/hr) and the FRR calculates the ratio of the number of false rejections (FRs) of keywords to the whole number of keyword occurrences. As a result, there may be FAs or FRs for a sample with a small loss, while a sample with a larger loss may not. Hence, sample importance can not be evaluated accurately just by the loss values. In this circumstance, the loss values-guided methods like focal loss can not work stably and sometimes may even impair the performance.

In this study, inspired by [16] and [18], we focus on using sample importance re-weighting to overcome data imbalance in the CTC-based KWS system and propose a number-of-errors guided re-weighted CTC loss (NER-CTCL). The keyword searching is incorporated into the training stage. And then, the difficulty or importance of samples is evaluated by the number of actual detection errors instead of the loss values. In other words, we concentrate on the FAs and FRs of a sample in keyword searching during training. The more FAs and FRs produced in a sample, the higher weight it will have. It helps to focus the training on hard examples quickly and automatically down-weights the contribution of easy examples. In fact, it also alleviates the inconsistency between the training objective and the metric mentioned above to a certain extent, as this helps training to optimize the performance metric, rather than to make it just a maximum likelihood learning process. Experiments compare NER-CTCL with standard CTC loss (S-CTCL) and focal CTC loss (F-CTCL) [24] that combines CTC loss and focal loss to handle the data imbalance. At an FAR of 0.5 fa/kw/hr, NER-CTCL achieves 16%–38% relative reductions in FRR against the S-CTCL. And it shows superiority compared to F-CTCL as well.

II. Related Work

1. Standard CTC loss

CTC is a popular method for sequence learning. It enables the E2E model training with no predefined alignment information required. The key idea of CTC is to introduce a blank symbol as an additional label to the label set and allow repetition of labels or blank across frames.

Given an input sequence $\boldsymbol{x}$ of length T, CTC trains the model to maximize the probability $p(l|\boldsymbol{x})$ for the corresponding target label sequence l of length $U\ (\leq T)$. Denoting the concatenation of observed labels at all time-steps as a path π. And a many-to-one mapping function $\mathcal{B}$ is defined to access the relationship between path π and target sequence l. Then, CTC represents the conditional probability as a summation of probabilities of all feasible paths as follows:

$$p(l|\boldsymbol{x}) = \sum_{\pi \in \mathcal{B}^{-1}(l)} p(\pi|\boldsymbol{x}) \tag{1}$$

Now, the CTC loss (i.e., S-CTCL) can be calculated as the negative logarithm probability of the ground truth as follows:

$$L_{\text{CTC}} = -\ln p(l|\boldsymbol{x}) \tag{2}$$

During training, the model is trained to minimize L_{CTC}.

2. Focal CTC loss

To alleviate the data imbalance, reference [24] com-

bined the CTC loss with focal loss and proposed F-CTCL, which is defined as follows:

$$L_{\text{(F)-CTC}}(l|\boldsymbol{x}) = -\alpha_t(1-p(l|\boldsymbol{x}))^{\gamma}\ln(p(l|\boldsymbol{x})) \tag{3}$$

where α_t ($\alpha_t \in [0,1]$) and γ (≥ 0) are two hyper-parameters that need to be tuned manually. This is practically a method of sample importance weighting, and it uses the loss values to evaluate the importance of samples. When loss value is large, i.e., $p(l|\boldsymbol{x})$ is small, the modulating factor $(1-p(l|\boldsymbol{x}))^{\gamma}$ is near 1 and the loss is unaffected; as $p(l|\boldsymbol{x}) \to 1$, the factor goes to 0 and the loss is down-weighted.

III. Methodology

In this paper, we also consider using sample importance re-weighting to overcome data imbalance in the CTC-based KWS system. Formally, the loss function of a sample $\boldsymbol{x}$ with target sequence l can be re-weighted as follows in sample importance re-weighting:

$$L_{\text{RE}} = -W \cdot \ln(p(l|\boldsymbol{x})) \tag{4}$$

where W is a weighting term, which represents sample importance.

As mentioned above, loss values-guided methods, such as focal loss [21], have been applied to alleviate the class-imbalance issue in KWS, but these approaches can not work stably since the loss values are not a suitable measurement to accurately evaluate the importance of samples in the CTC-based KWS system. This is actually due to the inconsistency between the objective of training and the evaluation metric, which are typically FAR and FRR. Hence, we use the number of detection errors, including FAs and FRs, to evaluate the importance of samples.

Specifically, given a training sample $(\boldsymbol{x}_i, \boldsymbol{y}_i)$, where $\boldsymbol{y}_i$ is the corresponding ground truth label sequence, the weight of $\boldsymbol{x}_i$ is denoted as $w(i,t)$, where t represents the t-th epoch. Firstly, the weights of all training samples are initialized to 1. In the traditional CTC-based KWS works, such as [13]–[15], keyword searching is not required at training stage, it is served as an indispensable tool to get the final results during testing only. In this study, in addition to calculating the loss in the forward pass during training, the keyword searching is applied to the model outputs to count the number of FAs and FRs for each training utterance:

$$N_{\text{F}}(i,t) = N_{\text{FA}}(i,t) + N_{\text{FR}}(i,t) \tag{5}$$

where $N_{\text{FA}}(i,t)$ and $N_{\text{FR}}(i,t)$ represent the number of FAs and FRs of the sample $\boldsymbol{x}_i$ in t-th epoch, respectively. Then, the weight for $\boldsymbol{x}_i$ at the end of t-th epoch is updated by

$$w(i,t) = N \times \frac{w'(i,t)}{\sum_{i=1}^{N} w'(i,t)} \tag{6}$$

$$w'(i,t) = \alpha w(i,t-1) + \beta N_{\text{F}}(i,t) \times \frac{1}{N_{\text{total}}(i,t) \times t} \tag{7}$$

where N is the number of training samples and $N_{\text{total}}(i,t)$ is the total number of FAs and FRs produced from $\boldsymbol{x}_i$ up to t-th epoch.

It is worth pointing out that sample importance should not be considered only from the current epoch, but from a global perspective, that is, it needs to be evaluated in conjunction with the statistics of the previous epochs. The more epochs a sample has generated FAs and FRs during training, the more difficult it can be trained, and the more attention needs to be paid to it. Therefore, equation (7) is set as a cumulative process, i.e. a sample once causes FAs or FRs in a certain epoch, its weight will increase. Hyperparameters α (> 0) and β (≥ 0) are used to weight the contribution of the original weight and the contribution of the false. Noting that as the training progresses, the performance of the model will gradually improve, and the overall level of the loss values will decrease. In order to avoid the weight accumulating to an excessively high level, resulting in large fluctuations of the loss values at the late training stage, the magnitude of the weight variation needs to be gradually reduced. So there is an attenuation factor t on the right side of (7). At the beginning of training, it helps the model focus on hard examples quickly. Additionally, there are always some samples that cause FAs or FRs at the late training stage. They are noisy labels or outliers if the errors exist stably even when the model is converged [25]–[27]. If large weights are blindly given to these samples, the performance would be impaired. Using the attenuation factor t and $N_{\text{total}}(i,t)$ can suppress this phenomenon to a certain extent as t reduces the impact of detection errors on the weights in the late training period when most samples can be correctly recognized, and $N_{\text{total}}(i,t)$ further prevents outliers that always produce errors (i.e., $N_{\text{total}}(i,t)$ becomes larger and larger) from misleading the training. The normalization (i.e., equation (6)) can ensure that the sum of the weights of all samples remains N, it therefore implicitly carries a "forgetting mechanism": if there are no detection errors in a sample, its weight will decrease. It should be pointed out that the normalization is done after each epoch, and the normalized weights are applied to the losses from the corresponding samples in the next epoch.

Now, the weighted CTC loss of $\boldsymbol{x}_i$ in t-th epoch is as follows:

$$L_{\text{NER-CTC}}(i,t) = w(i,t) \times L_{\text{CTC}}(i,t) = -w(i,t) \times \ln p(\boldsymbol{y}_i|\boldsymbol{x}_i) \quad (8)$$

where $\ln p(\boldsymbol{y}_i|\boldsymbol{x}_i)$ is calculated according to (1). In this way, the training will be guided to focus on undertrained samples and "ignore" relatively overtrained samples. In fact, for negative samples, as reference [18] pointed out, FAs are mainly caused by a small number of samples having pronunciation similar to the keyword during decoding. A large amount of negative samples tends to be easy to train and the hard examples are usually positive [25]. Therefore, NER-CTCL can alleviate the imbalance naturally and allows us to efficiently utilize all available data since it can automatically reduce the loss contribution from the majority of easy negative samples.

IV. Experiments

1. Dataset and experimental settings

The experiments are conducted on two public available Mandarin Chinese corpora: AISHELL-1 [28] and AISHELL-2 [29]. In this paper, only a part of the data extracted from the two corpora forms the experimental datasets. Detailed dataset statistics can be found in Table 1 and the specific keywords lists are shown in Table 2.

It should be pointed out that Table 1 only describes the statistics of positive samples, each of which contains at least one keyword consisting of 2 or 3 Chinese characters. For any special task in the experiments, the negative samples are added to form the final datasets according to the necessity of the task.

Table 1. Dataset statistics for positive samples

Dataset	Training set (75%)		Development set (10%)		Test set (15%)	
	#utters	#hrs	#utters	#hrs	#utters	#hrs
A2KW4	9,425	11.94	1,257	1.61	1,883	2.36
A2KW5	11,336	13.84	1,511	1.85	2,266	2.76
A2KW9	20,756	24.27	2,769	3.23	4,146	4.83
A2KW12	24,932	28.84	3,322	3.84	4,981	5.79
A2KW20	41,396	53.08	5,519	7.08	8,268	10.64
A1KW8	17,121	23.26	2,281	3.08	3,420	4.63

Note: "A2KW4" means 4 keywords need to be spotted and the dataset is sampled from AISHELL-2. The rest are similar ("A1" represents AISHELL-1). "utters" and "hrs" stand for utterances and hours, respectively.

Table 2. Keywords list

Dataset	Keywords
A2KW4	"shi2 jian1 (时间)", "shi4 jie4 (世界)", "shou3 ji1 (手机)", "shu4 ju4 (数据)"
A2KW5	"yin1 yue4 (音乐)", "ge1 qu3 (歌曲)", "wei4 shi4 (卫视)", "hu4 lian2 wang3 (互联网)", "ji1 qi4 ren2 (机器人)"
A2KW9	"gong1 zuo4 (工作)", "yu2 le4 (娱乐)", "chan3 pin3 (产品)", "cheng2 shi4 (城市)", "guo2 jia1 (国家)", "guo2 ji4 (国际)", "jing4 hua4 qi4 (净化器)", "fang2 di4 chan3 (房地产)", "dian4 shi4 ju4 (电视剧)"
A2KW12	"zhong1 yang1 (中央)", "xin4 xi1 (信息)", "ji4 shu4 (技术)", "zheng4 fu3 (政府)", "fu2 wu4 (服务)", "ji1 gou4 (机构)", "bi3 sai4 (比赛)", "dian4 ying3 (电影)", "guan3 li3 (管理)", "wen4 ti2 (问题)", "ji1 qi4 ren2 (机器人)", "jing4 hua4 qi4 (净化器)"
A2KW20	"shang4 hai3 (上海)", "ye4 wu4 (业务)", "zhong1 guo2 (中国)", "jiao1 yi4 (交易)", "chan3 pin3 (产品)", "jia4 ge2 (价格)", "qi3 ye4 (企业)", "gong1 si1 (公司)", "fa1 zhan3 (发展)", "fa1 bu4 (发布)", "cheng2 shi4 (城市)", "tu3 di4 (土地)", "ping2 guo3 (苹果)", "xiao1 shou4 (销售)", "xiang4 mu4 (项目)", "qi4 che1 (汽车)", "jian4 she4 (建设)", "dian4 shi4 ju4 (电视剧)", "hu4 lian2 wang3 (互联网)", "fang2 di4 chan3 (房地产)"
A1KW8	"ji4 zhe3 (记者)", "qi3 ye4 (企业)", "bai3 fen1 (百分)", "bei3 jing1 (北京)", "cheng2 shi4 (城市)", "shi4 chang3 (市场)", "zhong1 guo2 (中国)", "gong1 si1 (公司)"

In this study, we follow our previous work [15] to set the experiments. 80-dimensional log Mel spectrogram features with 25 ms frame length and 10 ms frame shift are extracted as model training inputs. The neural network for the KWS system is CRNN. For CNNs, there are three two-dimensional convolutional layers with the number of filters and kernel size as follows: [16, (3, 3)], [32, (3, 3)], and [32, (3, 3)]. The first and third convolutional layers are followed by a max-pooling layer, the size of which is 2×2 with a stride of 2×2. For RNNs, a two-layer BiGRU, each of which has 256 memory cells, is applied to model the sequential information. The Adam optimizer with an initial learning rate of 0.001, which decays every 10,000 steps with a base of 0.7, is chosen to train the model. The size of the mini-batch is 64. The development set are evaluated once every two epochs. When the performance of the development set has not improved after several epochs, an early stopping will be used to avoid overfitting.

2. Experimental results

As a contrast, the results of S-CTCL and F-CTCL [24] are also reported in this study. The hyperparameters α_t and γ in F-CTCL are tuned with the scheme used in [24].

As for NER-CTCL, different values of hyperparameters α and β in (7) were tested on "E^1-A2KW5" firstly, where 10 times negative samples are added to the "A2KW5" in the training set and 2 times in the test set. Here, "E^*-A2KW5" where "E" is the abbreviation of "extended" represents the extended version of "A2KW5", and the superscript stands for different versions (if exists); the same below. In the tuning process, the proportional relationship between α and β is mainly concerned because of the weights normalization (i.e., equation (6)). Therefore, the adjustment of hyperparameter can be easily converted to fix one parameter and then tune another one. The FRR results under different values of β with α being set to 1 are listed in Table 3 when FAR is fixed at 0.5 fa/kw/hr. It should be noted that NER-CTCL degenerates to S-CTCL when $\alpha = 1$ and $\beta = 0$. From Table 3, the advantages of NER-CTCL over S-CTCL can be seen. On the one hand, this advantage becomes more evident as β increases. According to (7), combined with the previous statistics of detection errors, the training will focus more on the samples which are hard to train for the current model when β increases. As the performance of the model gradually gets better during training, this evaluation for the "difficulty" of samples becomes more reliable to a certain extent. So when β is relatively large, the training makes better use of hard examples, and therefore better performance is obtained. Based on this, $\beta = 20$ was tried. However, it did not work. In this case, the FRR is higher than when $\alpha = 1$ and $\beta = 1$. On the other hand, the performance gradually deteriorates when β becomes smaller. In this situation, the weight of a sample is mainly dominated by accumulated errors. Because the performance of the model which is trained in the early training stage is relatively poor, the evaluation of the "difficulty" of the sample is relatively inaccurate; moreover, the inhibitory effect of $N_{\text{total}}(i,t)$ in the second term of (7) on outliers is greatly weakened since β is very small, so the training is affected by outliers and some "fake" hard examples, and the performance of the model decreases accordingly.

Table 3. FRR results on E^1-A2KW5 under different values of β with α being set to 1 when FAR is fixed at 0.5 fa/kw/hr

β	0	0.2	0.5	1	2	5	20
FRR	9.73	8.82	7.47	5.99	5.71	5.53	6.23

Note: The ratio of the number of positive samples and that of negative samples is 1:10 in the training set, and the ratio is 1:2 in the test set.

From Table 3, it can be seen that system performance is gradually getting better as β increases when β is less than 5. However, $\alpha = 1$, $\beta = 20$ performs worse than $\alpha = 1$, $\beta = 1$. Meanwhile, it shows that $\beta = 1$ is a "key point" in the view of the change of FRR vs. β:α, the decrease of FRR when β:α =1:2 becomes β:α =1:1 is much larger than that when β:α =1:1 changes to β:α =2:1. On the other hand, this paper focuses on the effect of the errors on training, which is an uncertain factor; for example, the errors may result from noise when the dataset is deteriorated by noise. Moreover, although $\alpha = 1$, $\beta = 1$ is not the optimal setting in the experiments in Table 3, its performance is much better than S-CTCL. Therefore, we think $\alpha = 1$, $\beta = 1$ is a reasonable choice based on these considerations. The following experiments will show that only using $\alpha = 1$, $\beta = 1$ without too many parameters tuning, NER-CTCL can also continuously show its superiority compared to S-CTCL and F-CTCL, which not only verifies the effectiveness but also demonstrates the robustness and ease of use of the proposed method. In other words, the performance of the KWS system can be improved without too much adjustment for hyperparameters.

The detection error trade-off (DET) curves comparing the performance of the systems that employ S-CTCL, F-CTCL, and NER-CTCL are shown in Fig.1. For visibility, only the two curves with the best results of F-CTCL are displayed. More detailed results can be seen in Table 4. It should be pointed out that the ratio of the number of positive and the number of negative samples is 1:2 in the test sets, while the ratio in the training sets is 1:10. In fact, for a certain keyword, the samples for other keywords are negative in the multi-keyword KWS system. Therefore, the imbalance is more serious. For example, after adding negative samples to the training set, the imbalance can be more serious than 1:40 as for "E-A2KW4". And at the character level, the actual imbalance can be even more severe as the non-keyword in the positive samples are also negatives.

From Fig.1 and Table 4, the effectiveness of NER-CTCL can be seen. As for "E-A2KW4", when the FAR is 0.5 fa/kw/hr, NER-CTCL yields a lower FRR than S-CTCL with a 1.81% absolute reduction, which relatively decreases by 16.45%. And for "E^1-A2KW5", "E-A2KW9", "E-A2KW12", "E-A2KW20", and "E-A1KW8", the absolute reductions are 3.74%, 1.59%, 1.54%, 0.96%, and 1.47%, a relative reduction of 38.44%, 20.08%, 27.80%, 28.49%, and 30.37%, respectively. Moreover, F-CTC also shows some superiorities

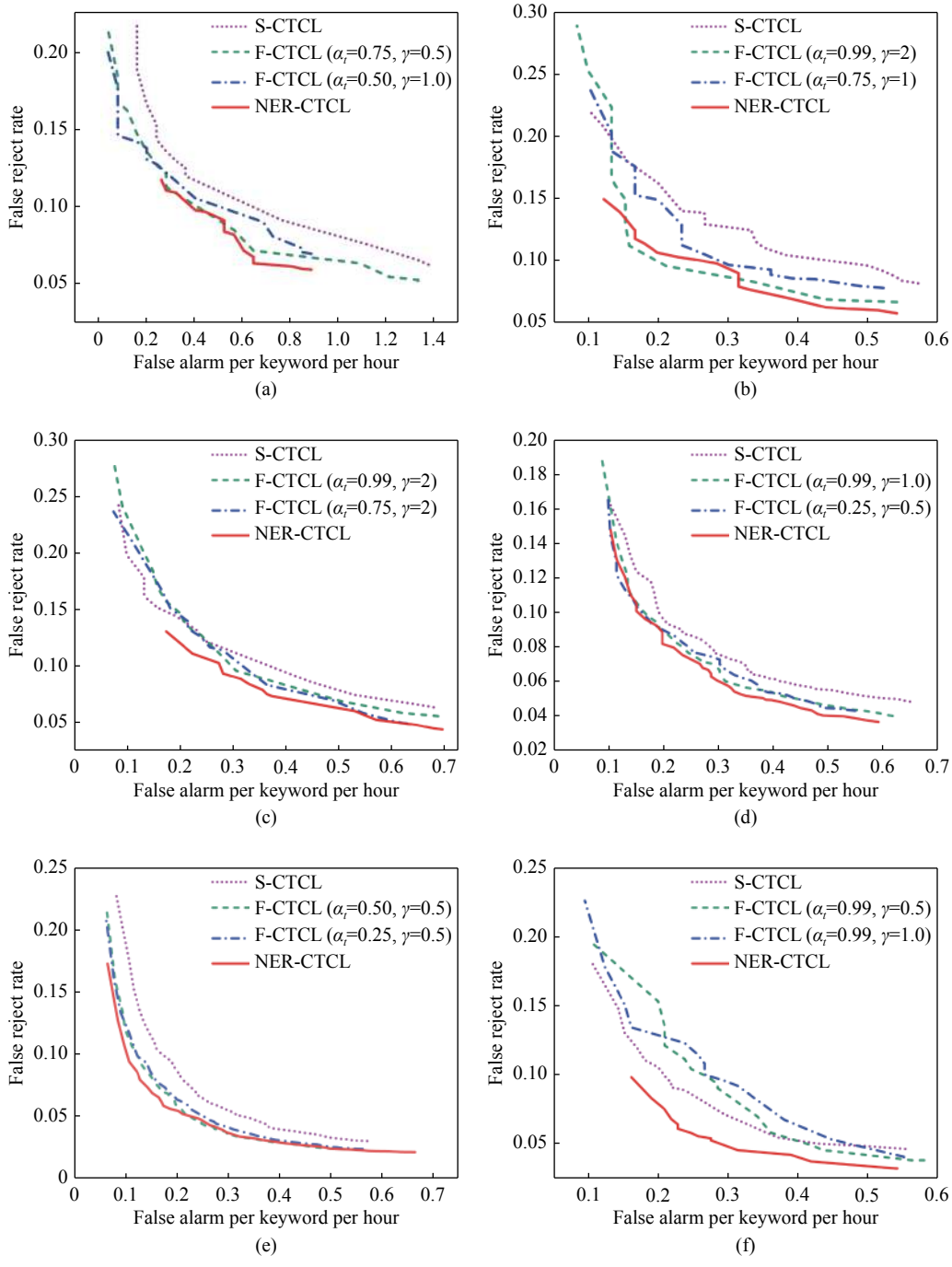

Fig. 1. DET curves comparing performance of the systems that employ S-CTCL, F-CTCL, and NER-CTCL for (a) "E-A2KW4"; (b) "E[1]-A2KW5"; (c) "E-A2KW9"; (d) "E-A2KW12"; (e) "E-A2KW20"; and (f) "E-A1KW8".

compared to S-CTCL and sometimes even gets the best result. However, the improvement is not easy to get and is very unstable. As can be seen from Table 4, some bad choices of α_t and γ impair the performance. In fact, as for "E-A1KW8", despite many choices being tried, F-CTCL does not work. And the optimal choice is not the same in different cases. In other words, if F-CTCL is used to improve performance expectedly, there is a vast parameter space need to search, while NER-CTCL achieves or closely achieves the best result just in a nearly effortless way. Besides, as for "E-A2KW4", it can also be found from Fig.1(a) that for S-CTCL and F-CTCL, the range of FAR is overall wide, which is actually because the pronunciation of the four keywords is similar ("sh- j-"), and NER-CTCL reduces the range a lot.

To further verify the effectiveness of NER-CTCL, different numbers of negative samples are added to the "A2KW5" to observe the performance of NER-CTCL on different degrees of imbalanced training sets, the ra-

Table 4. FRR results among S-CTCL, F-CTCL, and NER-CTCL on different datasets with FAR fixed at 0.5 fa/kw/hr

Loss function	α_t	γ	FRR					
			E-A2KW4	E[1]-A2KW5	E-A2KW9	E-A1KW12	E-A1KW20	E-A1KW8
S-CTCL	–	–	11.00	9.73	7.92	5.54	3.37	4.84
F-CTCL	0.99	0.5	10.96	8.98	7.14	5.39	2.80	4.48
	0.99	1	13.23	10.11	7.93	4.61	2.92	4.67
	0.99	2	9.94	6.78	7.01	5.63	2.69	5.71
	0.75	0.5	**9.15**	8.54	7.80	5.32	3.52	5.20
	0.75	1	10.61	7.95	9.17	5.71	3.23	5.86
	0.75	2	9.67	9.69	6.81	4.80	2.95	5.43
	0.5	0.5	10.19	9.07	7.86	5.72	**2.39**	5.26
	0.5	1	10.00	10.61	9.45	4.83	2.88	5.47
	0.5	2	10.49	9.32	8.99	4.67	2.59	5.52
	0.25	0.5	12.09	12.37	8.70	4.46	2.53	5.71
	0.25	1	11.08	10.18	7.95	4.70	2.51	5.48
	0.25	2	9.40	9.03	7.65	4.69	2.92	6.85
NER-CTCL	–	–	9.19	**5.99**	**6.33**	**4.00**	2.41	**3.37**

tio is fixed to 1:2 in test set. The experimental results are shown in Fig.2. Combined with Fig.1, it can be seen that NER-CTCL keeps its superiority in different situations where the imbalance ratios are different. Specifically, at an FAR of 0.3 fa/kw/hr, when the imbalance ratios (positive:negative) are 1:5, 1:10, 1:15, 1:20, and 1:25, compared to S-CTCL, using NER-CTCL reduces the FRR from 13.04%, 12.71%, 10.66%, 11.74%, and 11.48% to 10.94%, 9.49%, 8.36%, 8.80%, and 7.44%, respectively; the relative reductions are 16.10%, 25.33%, 21.58%, 25.04%, and 35.19%, respectively. It can be found from Fig.1(b) and Fig.2 that as the number of negative samples increases, FAR decreases to a certain extent, and the overall performance of the KWS system is improved. But this improvement is not unlimited. When there are too many negative samples, the performance of S-CTCL will decrease obviously due to the data imbalance. And NER-CTCL can effectively alleviate this problem. In fact, when the training set becomes more and more imbalanced, the relative performance improvement brought by NER-CTCL seems to be more and more.

Although NER-CTCL has more computation in training than S-CTCL since additional keyword search-

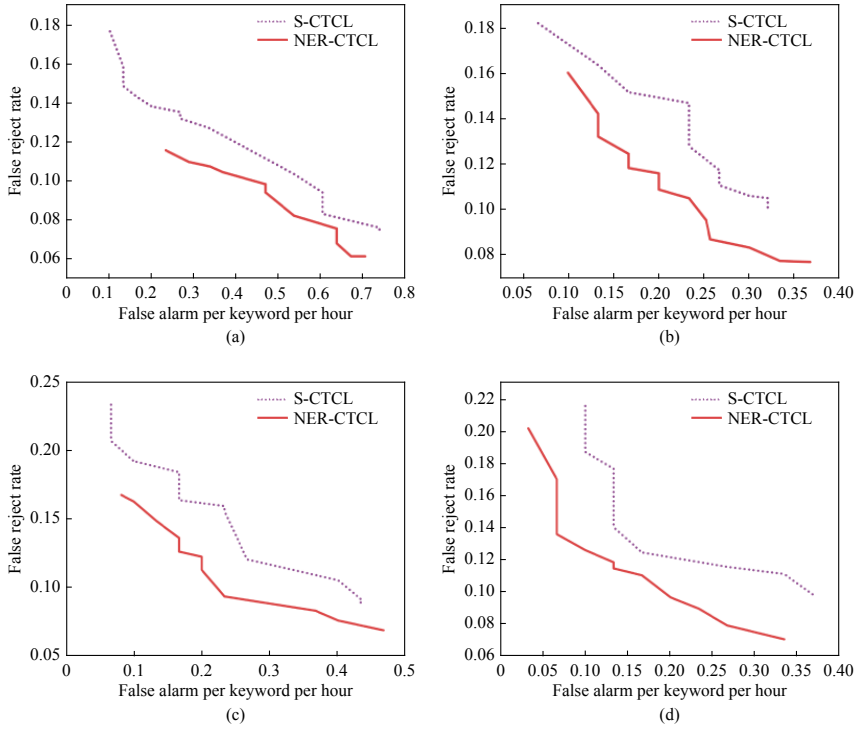

Fig. 2. DET curves on different degrees of imbalanced datasets. The negative samples are added to the training set at the ratios (positive:negative) of (a) 1:5; (b) 1:15; (c) 1:20; and (d) 1:25 in "A2KW5". The ratio is 1:2 in the test set.

ing is required to get the accumulated statistics about FRs and FAs, it actually does not increase the training time too much. In order to evaluate the performance of the proposed method better, a comparison of training time consumption is conducted for quantitative analysis. The results which are obtained by averaging the time cost of each epoch for the entire training stage are shown in Table 5, where the value in parentheses represents the standard deviation. For reliability, the machine is guaranteed to run in a basically consistent environment in the two experiments. It can be found that there is only a tiny extra time consumption when using NER-CTCL. Specifically, it only increases by 0.16 ms (i.e., 1.79%) per epoch for each sample on average.

Table 5. Comparison of training time consumption

Loss function	Training time (s/epoch)	Average (ms/utterance/epoch)
S-CTCL	994.73 (±2.02)	8.95
NER-CTCL	1,012.66 (±1.89)	9.11

Note: "E[1]-A2KW5" where 101,070 negative samples are added to the training set in "A2KW5" is used to count the training time, i.e. the total number of utterances is 111,177 in the training set.

Finally, we look into the training procedure to find how the NER-CTCL works. The weights of the samples are tracked during training. Taking "E-A2KW9" as an example, when 207,560 negative examples are used (i.e., the ratio of the number of positive and the number of negative samples is 1:10), until the end of the training, 17,578 (84.69%) positive samples have caused FAs or FRs during the training, resulting them having greater weights than those samples that have never produced detection errors.

In contrast, only 2,673 (1.29%) negative samples have triggered FAs; most negative samples actually own the smallest weight. Simultaneously, we observe the weights and find that at the start of training, the ratio of the sum of the weights of positive samples and the sum of the weights of negative samples is 1:10, it changes to about 1:5 at the end of the training, which alleviates the imbalance from the overall weights. In fact, the more important thing that NER-CTCL is effective is that it makes the training discriminative, i.e. those samples that deserve more attention are emphatically trained whether they are positive or negative samples.

V. Conclusions

This study explores using sample importance reweighting to handle the data imbalance of CTC-based KWS and proposes a novel re-weighted CTC loss NER-CTCL, which evaluates sample importance by the number of detection errors (including FAs and FRs). NER-CTCL weights the standard CTC loss to make the training focus on hard examples and down-weight the numerous easy examples. It is a simple and effective approach. Experiments verified the superiority of NER-CTCL compared to standard CTC loss, as well as focal CTC loss.

References

[1] J. Gao, J. Shao, Q. W. Zhao, *et al.*, "Efficient system combination for Chinese spoken term detection," *Chinese Journal of Electronics*, vol.19, no.3, pp.457–462, 2010.

[2] E. F. Huang, H. C. Wang, and F. K. Soong, "A fast algorithm for large vocabulary keyword spotting application," *IEEE Transactions on Speech and Audio Processing*, vol.2, no.3, pp.449–452, 1994.

[3] S. Tabibian, "A voice command detection system for aerospace applications," *International Journal of Speech Technology*, vol.20, no.4, pp.1049–1061, 2017.

[4] J. T. Foote, S. J. Young, G. J. F. Jones, *et al.*, "Unconstrained keyword spotting using phone lattices with application to spoken document retrieval," *Computer Speech & Language*, vol.11, no.3, pp.207–224, 1997.

[5] C. L. Zhu, Q. J. Kong, L. Zhou, *et al.*, "Sensitive keyword spotting for voice alarm systems," in *Proceedings of 2013 IEEE International Conference on Service Operations and Logistics, and Informatics*, Dongguan, China, pp.350–353, 2013.

[6] J. G. Wilpon, L. R. Rabiner, C. H. Lee, *et al.*, "Automatic recognition of keywords in unconstrained speech using hidden Markov models," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol.38, no.11, pp.1870–1878, 1990.

[7] M. C. Silaghi, "Spotting subsequences matching a HMM using the average observation probability criteria with application to keyword spotting," in *Proceedings of the 20th National Conference on Artificial Intelligence*, Pittsburgh, PA, USA, pp.1118–1123, 2005.

[8] V. Frinken, A. Fischer, R. Manmatha, *et al.*, "A novel word spotting method based on recurrent neural networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.34, no.2, pp.211–224, 2012.

[9] G. G. Chen, C. Parada, and G. Heigold, "Small-footprint keyword spotting using deep neural networks," in *Proceedings of 2014 IEEE International Conference on Acoustics, Speech and Signal Processing*, Florence, Italy, pp.4087–4091, 2014.

[10] T. N. Sainath and C. Parada, "Convolutional neural networks for small-footprint keyword spotting," in *Proceedings of the 16th Annual Conference of the International Speech Communication Association*, Dresden, Germany, pp.1478–1482, 2015.

[11] M. Sun, D. Snyder, Y. X. Gao, *et al.*, "Compressed time delay neural network for small-footprint keyword spotting," in *Proceedings of the 18th Annual Conference of the International Speech Communication Association*, Stockholm, Sweden, pp.3607–3611, 2017.

[12] A. Graves, S. Fernández, F. Gomez, *et al.*, "Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks," in *Proceedings of the 23rd International Conference on Machine Learning*, Pittsburgh, PA, USA, pp.369–376, 2006.

[13] M. Wöllmer, B. Schuller, and G. Rigoll, "Keyword spotting

exploiting long short-term memory," *Speech Communication*, vol.55, no.2, pp.252–265, 2013.
[14] Y. Bai, J. Y. Yi, H. Ni, *et al.*, "End-to-end keywords spotting based on connectionist temporal classification for mandarin," in *Proceedings of the 10th International Symposium on Chinese Spoken Language Processing*, Tianjin, China, pp.1–5, 2016.
[15] H. K. Yan, Q. H. He, and W. Xie, "CRNN-CTC based mandarin keywords spotting," in *Proceedings of 2020 IEEE International Conference on Acoustics, Speech and Signal Processing*, Barcelona, Spain, pp.7489–7493, 2020.
[16] B. Liu, S. Nie, Y. P. Zhang, *et al.*, "Loss and double-edge-triggered detector for robust small-footprint keyword spotting," in *Proceedings of 2019 IEEE International Conference on Acoustics, Speech and Signal Processing*, Brighton, UK, pp.6361–6365, 2019.
[17] J. Y. Hou, Y. Y. Shi, M. Ostendorf, *et al.*, "Mining effective negative training samples for keyword spotting," in *Proceedings of 2020 IEEE International Conference on Acoustics, Speech and Signal Processing*, Barcelona, Spain, pp.7444–7448, 2020.
[18] K. Zhang, Z. Y. Wu, D. D. Yuan, *et al.*, "Re-weighted interval loss for handling data imbalance problem of end-to-end keyword spotting," in *Proceedings of 21st Annual Conference of the International Speech Communication Association*, Shanghai, China, pp.2567–2571, 2020.
[19] C. Elkan, " The foundations of cost-sensitive learning," in *Proceedings of the 17th International Joint Conference on Artificial Intelligence*, Seattle, WA, USA, vol.17, pp.973–978, 2001.
[20] Y. Cui, M. L. Jia, T. Y. Lin, *et al.*, " Class-balanced loss based on effective number of samples," in *Proceedings of 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Long Beach, CA, USA, pp.9260–9269, 2019.
[21] T. Y. Lin, P. Goyal, R. Girshick, *et al.*, " Focal loss for dense object detection," in *Proceedings of 2017 IEEE International Conference on Computer Vision*, Venice, Italy, pp.2999–3007, 2017.
[22] S. Ben-David, D. Loker, N. Srebro, *et al.*, "Minimizing the misclassification error rate using a surrogate convex loss," in *Proceedings of the 29th International Conference on Machine Learning*, Edinburgh, UK, pp.83–90, 2012.
[23] Y. B. Zhou, C. M. Xiong, and R. Socher, "Improving end-to-end speech recognition with policy learning," in *Proceedings of 2018 IEEE International Conference on Acoustics, Speech and Signal Processing*, Calgary, Canada, pp.5819–5823, 2018.
[24] X. J. Feng, H. X. Yao, and S. P. Zhang, "Focal CTC loss for Chinese optical character recognition on unbalanced datasets," *Complexity*, vol.2019, article no.9345861, 2019.
[25] B. Y. Li, Y. Liu, and X. G. Wang, "Gradient harmonized single-stage detector," in *Proceedings of the 33rd AAAI Conference on Artificial Intelligence*, Honolulu, HI, USA, vol.33, pp.8577–8584, 2019.
[26] M. Toneva, A. Sordoni, R. T. des Combes, *et al.*, "An empirical study of example forgetting during deep neural network learning," in *Proceedings of the 7th International Conference on Learning Representations*, New Orleans, LA, USA, 2019.
[27] L. Jiang, Z. Y. Zhou, T. Leung, *et al.*, "MentorNet: Learning data-driven curriculum for very deep neural networks on corrupted labels," in *Proceedings of 35th International Conference on Machine Learning*, Stockholm, Sweden, pp.2304–2313, 2018.
[28] H. Bu, J. Y. Du, X. Y. Na, *et al.*, "AISHELL-1: An open-source mandarin speech corpus and a speech recognition baseline," in *Proceedings of 20th Conference of the Oriental Chapter of the International Coordinating Committee on Speech Databases and Speech I/O Systems and Assessment*, Seoul, Korea (South), pp.1–5, 2017.
[29] J. Y. Du, X. Y. Na, X. C. Liu, *et al.*, "AISHELL-2: Transforming mandarin ASR research into industrial scale," *arXiv preprint*, arXiv: 1808.10583, 2018.

LAN Xiaotian received the B.S. degree in communication engineering from Northeastern University, Shenyang, in 2019. He is currently pursuing the M.S. degree at South China University of Technology. His research interests include speech keyword spotting and automatic speech recognition.
(Email: xtianlan@163.com)

HE Qianhua (corresponding author) received the Ph.D. degree in communication engineering from South China University of Technology in 1993, where he is currently a Full Professor with the School of Electronic and Information Engineering, the B.S. degree in physics from Hunan Normal University in 1987, and the M.S. degree in medical instrument engineering from Xi'an Jiaotong University in 1990. From 2007 to 2008, he was with University of Washington in Seattle USA as a Visiting Scholar. From 1994 to 2001, he was with the Department of Computer Science, City University of Hong Kong, four times as a Research Assistant, a Senior Research Assistant, and a Research Fellow, respectively. His research interests include spoken term detection, audio event detection, speech coding, multimedia retrieval, and digital audio forensic. He is a Senior Member of IEEE. (Email: eeqhhe@scut.edu.cn)

YAN Haikang received the B.S. degree in communication engineering from Southwest Jiaotong University in 2018. He is currently pursuing the M.S. degree with the South China University of Technology (SCUT). His research interests include speech keyword spotting and audio signal processing.
(Email: haikangyan@163.com)

LI Yanxiong received the B.S. and M.S. degrees in electronic engineering from Hunan Normal University, Changsha, China, in 2003 and 2006, respectively, and Ph.D. degree in electronic engineering from SCUT, Guangzhou, China, in 2009. From 2008 to 2009, he worked as a Research Associate with the City University of Hong Kong. From 2013 to 2014, he worked as a Researcher with the University of Sheffield, UK. From Jul. to Aug. 2016, he worked as a Visiting Scholar with the Institute for Infocomm Research, Singapore. From Jul. to Oct. 2019, he worked as a Visiting Scholar with the Tampere University of Technology (TUT), Finland. He is now an Associate Professor with the School of Electronic and Information Engineering, SCUT. His research interests include audio signal processing and machine learning. (Email: eeyxli@scut.edu.cn)