SPARSE SELF-ATTENTION FOR SEMI-SUPERVISED SOUND EVENT DETECTION

Yadong Guan, Jiabin Xue, Guibin Zheng, Jiqing Han

School of Computer Science and Technology, Harbin Institute of Technology, Harbin, China

ABSTRACT

Self-attention mechanism has been widely employed in semi-supervised sound event detection (SS-SED). In self-attention, since dependencies between pairwise features at all moments are captured, the irrelevant features of different classes of sounds and background sounds at other moments are inevitably mixed in the current embedding when self-attention performs weighted summation. These irrelevant features will weaken the ability of the aggregated embedding to describe sound events. In this paper, we propose a sparse self-attention mechanism to alleviate the impact. Specifically, the Sparsemax function is introduced for attention weights normalization, which uses Euclidean projection to project attention weights onto a probability simplex. After the normalization, the attention weights of the irrelevant features are projected onto the boundary of the simplex and then removed. Furthermore, to solve the excessive sparsity problem of the Sparsemax, we further propose the Sparsemax with adjustable sparsity. Experimental results demonstrate the effectiveness of the proposed method.

Index Terms— Semi-supervised Sound Event Detection, Sparse Self-attention, Sparsemax

1. INTRODUCTION

Sound event detection (SED) is the task of recognizing sound events and locating them temporally in audio recordings. As an essential part of ambient sound analysis, SED can be used for surveillance [1], multimedia indexing [2], smart cities [3–5], and homes [6]. Recently, the Detection and Classification of Acoustic Scenes and Events (DCASE) challenges [7] further promote the research on SED. Due to the lack of data with frame-level annotations, semi-supervised sound event detection (SS-SED) [8–11] has received extensive attention.

In general, sound events in audio clips have the following properties. Firstly, sound events usually do not last throughout audio clips and sometimes only account for a small part. Secondly, there may be long-lasting background sounds and various sound events in audio clips. When humans detect current sound events, they do not often rely on features of different classes of sound events at other moments. However, for the popular SS-SED model based on self-attention [12–14], the attention weights are positive after being normalized by the Softmax function. This means that self-attention captures dependencies between pairwise features at all moments. As a result, the irrelevant features of different classes of sound events or background sounds at other moments are inevitably mixed in the current embedding when self-attention performs weighted summation. These irrelevant features will weaken the ability of the aggregated embedding to describe sound events and impair its discrimination [9, 15]. The situation is severe for short-duration sound events because irrelevant features may be dominant due to their higher ratio of duration. Hence, it is necessary to find and remove attention on irrelevant features. However, for SS-SED, the lack of frame-level labels makes it difficult to implement.

Generally speaking, attention weights between features of different classes of sound events are smaller than those between features of the same class of events. Thus, a simple way to remove attention on irrelevant features is to set small attention weights to zero. To this end, we propose a new sparse self-attention model by introducing the Sparsemax [16] for attention weights normalization. The Sparsemax function works by projecting attention weights onto a probability simplex. In the projection process, smaller attention weights will fall on the boundary of the simplex and become zero. In addition, the Sparsemax function has the problem of excessive sparsity; i.e., some large attention weights may also be projected onto the boundary of the simplex, which is unreasonable. To solve this problem, we propose the Sparsemax with adjustable sparsity by introducing a hyperparameter to control its sparsity.

2. SPARSE SELF-ATTENTION

2.1. Problem Formulation

CNN+Transformer [14] model is adopted as the backbone, mainly containing CNN [17] for extracting local features and the Transformer encoder for capturing temporal features. After the Transformer encoder, a fully connected layer is used for frame-level classification. The input log-mel spectrogram feature is denoted as $\boldsymbol{X} \in \mathbb{R}^{T \times F}$, where T and F represent the total number of frames and mel-bins. $\boldsymbol{H} \in \mathbb{R}^{T' \times D}$ is obtained through CNN, where T' and D are the length of feature

This research was supported by the National Natural Science Foundation of China under Grant U1736210.

sequence and feature dimension. Next, query($\boldsymbol{Q}$), key($\boldsymbol{K}$) and value($\boldsymbol{V}$) matrices can be obtained in the Transformer encoder as in [14]. The matrix of self-attention weights is formulated as:

$$\boldsymbol{A} = \frac{\boldsymbol{Q}\boldsymbol{K}^{\mathrm{T}}}{\sqrt{d}} \tag{1}$$

where $\boldsymbol{A} \in \mathbb{R}^{T' \times T'}$, d is the dimension of the key. The attention weights $A_{t,i}$ in $\boldsymbol{A}$ are normalized using the Softmax:

$$A'_{t,i} = \frac{\exp\left(A_{t,i}\right)}{\sum_{j=1}^{T'} \exp\left(A_{t,j}\right)}, t, i \in \{1, ..., T'\} \tag{2}$$

After the normalization, the attention weights are strictly positive. Then the attention weights in $\boldsymbol{A'}$ are used to perform weighted summation on the vectors in $\boldsymbol{V}$:

$$\boldsymbol{E} = \boldsymbol{V}\boldsymbol{A'}^{\mathrm{T}} \tag{3}$$

In general, a single sound event does not last throughout an audio clip, and there may be other classes of sound events and background sounds. Moreover, the features of different classes of sound events are usually irrelevant. Thus, in the weighted summation process, the global attention will lead to a mix of irrelevant features. The situation is severe for short-duration sound events because irrelevant features may account for a higher proportion in time. In conclusion, the traditional Softmax normalization function is not suitable for the self-attention-based SED model.

2.2. Sparse Self-attention

The attention weight measures the dependency between features of two moments. The dependencies between features of the same class of sound events are usually stronger, while the dependencies between features of different classes of sound events are weaker. In general, the strength of feature dependency is expressed by the value of attention weights. A larger weight means a stronger dependency. Thus, as a whole, attention weights between features of different classes of sound events are smaller than those between features of the same class of events. Therefore, the feature dependencies between different classes of sound events can be removed by resetting smaller weights to zero.

Based on the above analysis, we propose a new sparse self-attention model by introducing the Sparsemax for attention weights normalization. It can be achieved by solving the following optimization problem with a simplex constraint.

$$\mathrm{Sparsemax}\left(\boldsymbol{A}_{t,:}\right) = \underset{\boldsymbol{p} \in \Delta^{T'-1}}{\arg\min} \frac{1}{2} \left\| \boldsymbol{A}_{t,:}^{\mathrm{T}} - \boldsymbol{p} \right\|_2^2 \tag{4}$$

where $\Delta^{T'-1} = \{\boldsymbol{p} \in \mathbb{R}^{T'} | \boldsymbol{p} > \boldsymbol{0}, \sum_{i=1}^{T'} p_i = 1\}$ is a probability simplex, $\boldsymbol{A}_{t,:} = [a_{t,1}, a_{t,2}, ..., a_{t,T'}]$, $\mathrm{Sparsemax}\left(\cdot\right)$ is the Sparsemax function. Intuitively, Eq. (4) returns the Euclidean projection of the attention weight vector $\boldsymbol{A}_{t,:}$ onto the probability simplex $\Delta^{T'-1}$. In the projection process, smaller attention weights will fall on the boundary of the simplex and become zero. According to [18], the closed-form solution of the Eq. (4) is as follows:

$$a'_{t,i} = \left[a_{t,i} - \tau(\boldsymbol{A}_{t,:})\right]_+ \tag{5}$$

where $\left[\cdot\right]_+$ is $\max(\cdot, 0)$, $\tau(A_{t,:})$ is a dynamic threshold, and $\tau : \mathbb{R}^{T'} \rightarrow \mathbb{R}$ is the function that satifies:

$$\sum\nolimits_j \left[a_{t,j} - \tau(\boldsymbol{A}_{t,:})\right]_+ = 1 \tag{6}$$

The overall procedure of sparse self-attention is summarized in Algorithm 1. Firstly, the attention matrix $\boldsymbol{A}$ is calculated. Then for each $\boldsymbol{A}_{t,:}$, we figure out the intermediate variable k_t and the threshold τ_t. The dynamic threshold is determined by the overall distribution of attention weights. Those attention weights below this threshold will become zero. Finally, the normalized attention matrix $\boldsymbol{A'}$ and the output feature matrix $\boldsymbol{E}$ can be obtained. The time complexity of normalizing the attention weights using the Sparsemax is $O(T'^2 \log T')$, which is slightly larger than $O(T'^2)$ of the Softmax.

Algorithm 1 Sparse Self-attention for SED

Input: query $\boldsymbol{Q}$, key $\boldsymbol{K}$, value $\boldsymbol{V}$

$\boldsymbol{A} = \frac{\boldsymbol{Q}\boldsymbol{K}^{\mathrm{T}}}{\sqrt{d}}$
for $t = 1, 2, ..., T'$ **do**
  descending sort $\boldsymbol{A}_{t,:}$
  find $k_t = \max\left\{k \in [T'] \middle| 1 + kA_{t,k} > \sum_{j \leq k} A_{t,j}\right\}$
  $\tau(\boldsymbol{A}_{t,:}) = \frac{(\sum_{j \leq k_t} A_{t,j}) - 1}{k_t}$
  for $j = 1, 2, ..., T'$ **do**
    $A'_{t,j} = \left[A_{t,j} - \tau(\boldsymbol{A}_{t,:})\right]_+$
  end for
end for
$\boldsymbol{E} = \boldsymbol{V}\boldsymbol{A'}^{\mathrm{T}}$

Output: $\boldsymbol{E}$

2.3. Sparsemax with Adjustable Sparsity

When using the Sparsemax to normalize attention weights, some attention weights, although not very small, may also be projected on the boundary of the probability simplex and become zero. For example, $[0.5, 4, 5]$ becomes $[0, 0, 1]$ after the Sparsemax normalization. As can be seen from this example, the Sparsemax may lose some useful information. The following proposition will quantify this problem.

Proposition 1 *Let $\boldsymbol{b} \in \mathbb{R}^n$ be the vector to be normalized, sort the elements in $\boldsymbol{b}$ in descending order, $\boldsymbol{c} = Sparsemax\left(\boldsymbol{b}\right)$, k is the number of non-zero elements in $\boldsymbol{c}$, $\forall j : 1 \leq j \leq n$, if $\exists j : b_j - b_{j+1} \geq 1$, then $j > k$.*

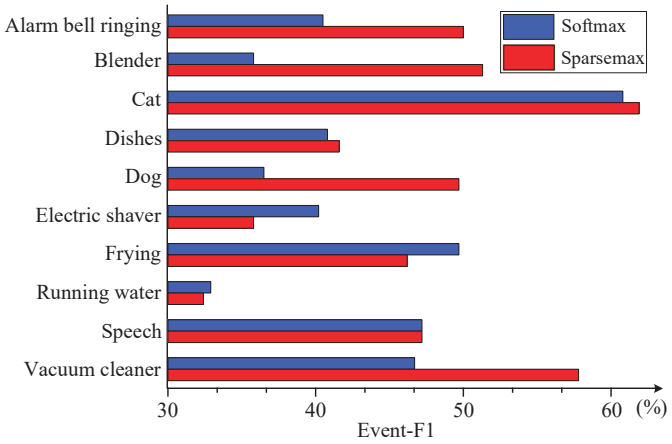

Fig. 1. Class-wise performance comparison.

Proof: The proof and experimental code are available[1].

Prop.1 states that as long as the difference between two adjacent attention weights in a descending vector is greater than or equal to one, even if they are both large, after the Sparsemax function, the smaller weight will become zero. Thus, this excessive sparsity may cause the loss of some important information. For ease of description, we define one as the sparsity of the Sparsemax function.

To solve the problem, we propose the Sparsemax with adjustable sparsity, which reduces attention weights by $\lambda(\lambda > 1)$ times before normalization, as shown below.

$$Sparsemax\left(\boldsymbol{A}_{t,:}\right) = \underset{\boldsymbol{p}\in\Delta^{T'-1}}{\arg\min} \frac{1}{2}\left\| \frac{\boldsymbol{A}_{t,:}^{\mathrm{T}}}{\lambda} - \boldsymbol{p} \right\|_2^2 \quad (7)$$

In this way, the sparsity of the improved Sparsemax becomes λ. Only when the difference between two adjacent attention weights in the descending vector is greater than or equal to λ, will the smaller weight after the Sparsemax normalization become zero.

3. EXPERIMENTAL SETUP

We evaluated our method on the DESED [19] dataset. This dataset contains ten classes of common sounds in the domestic environment. We resampled all the audio clips to 16kHz. We used 128-dimensional log-mel spectrograms as input features. The window size and hop length were 2048 and 255, respectively. MixUp [20] and time-shifting were used for data augmentation.

The mean teacher model [17] was adopted for semi-supervised learning. For the student model, we used the same CNN as in [17], with the Transformer encoder cascading behind it. We adopted the Transformer encoder implementation in PyTorch with 16 heads and 512 feed-forward units.

[1] https://github.com/guanyadong/ssa_sed

Table 1. Performance of the SSA under different sparsity

	baseline	λ (SSA)					
		1.0	1.1	1.2	1.3	1.4	1.5
F1(%)	43.1	46.8	45.1	46.1	**47.6**	46.6	45.8

The normalization function for self-attention in the baseline was the Softmax function. The dropout rate was 0.2. The loss functions were the binary cross entropy loss and the consistency loss [17]. The learning rate was 0.001 with warm-up scheduling. We trained the models for 500 epochs. For post-processing, we set different lengths of median filter windows for each class of sound events as in [17]. The macro event-F1 [21] and PSDS [22] were adopted as the evaluation metrics.

4. EXPERIMENTAL RESULTS AND ANALYSIS

4.1. Selection of Optimal Sparsity for SSA

We first compared the performance of the baseline and the SSA under different sparsity λ, as shown in Table 1. Event-F1 of the SSA is higher than the baseline, regardless of the λ. Moreover, when the sparsity λ is taken as 1.3, the F1-score reaches the highest 47.6%, higher than the 46.8% before improvement. The improvement proves the effectiveness of the Sparsemax with adjustable sparsity.

Table 2. SS-SED performance on the DESED dataset

Model	**Event-F1(%)**	**PSDS(%)**
CRNN [23]	45.1	65.8
CRNN ensemble [23]	46.4	-
FP-CRNN [23]	44.9	**66.9**
CNN+Transformer [13]	46.0	64.3
baseline	43.1	64.1
Our method (SSA)	**47.6**	66.7

4.2. Performance Comparison of Different Models

The performance of different models is shown in Table 2. Event-F1 and PSDS of the SSA model are 47.6% and 66.7%, higher than the baseline (43.1% and 64.1%). As can be seen, the baseline performance is lower than that of the CRNN (45.1% F1, 65.8% PSDS) and the FP-CRNN (44.9% F1, 66.9% PSDS), while the SSA model outperforms them. Especially, the SSA model surpasses the ensemble of the CRNN (46.4% F1). This shows that in the SS-SED task, the temporal modeling capacity of the sparse self-attention exceeds that of the RNN. The SSA model surpasses the single transformer-based model of the first place in the DCASE 2020 task4 (46.0% F1, 64.3% PSDS).

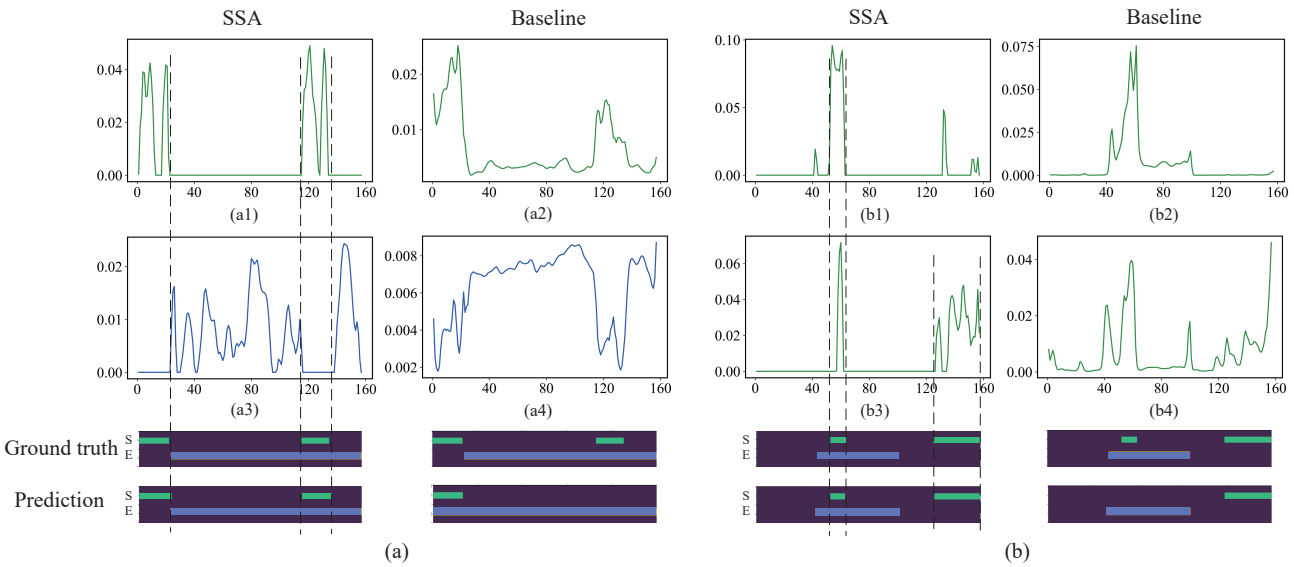

Fig. 2. Distribution of normalized attention weights in the SSA and the baseline. Subgraphs (a) and (b) are the results of two audio clips. (a1) and (a3) are distributions of attention weights in the SSA, and (a2) and (a4) are distributions in the baseline. The horizontal and vertical axis represent time and attention weights, respectively. (a1) and (a3) come from different self-attention heads. Attention weights in each subgraph represent the similarity between the features of one moment and those of all moments. The color of the curve is related to the label. The bottom figures show the label of ground truth and prediction, where S and E represent speech and electric shaver, respectively.

4.3. Class-wise Performance Comparison

The class-wise performance of the baseline and the SSA model is shown in Fig.1. The SSA model outperforms the baseline for most short-duration events, e.g., alarm bell ringing, blender, cat, dishes, and dog, especially for alarm bell ringing, blender, and dog. It can be easily explained by the fact that these events are short, while irrelevant features of different classes of sounds or background sounds have a relatively long duration. And sparse self-attention can significantly reduce the proportion of irrelevant features in the weighted summation process. However, the SSA performs slightly worse than the baseline for long-duration events (frying, electric shaver, and running water). The reason may attribute to poor discrimination of these sound events, which requires long-term features for judgment, and the sparse self-attention may lead to the loss of some valuable features.

4.4. Visualization of Self-attention Weights

To illustrate the benefits of the SSA intuitively, we visualized the attention weights in the SSA and baseline model, as shown in Fig.2. It can be seen that the prediction of the SSA is correct while false positive and false negative errors occur in the prediction of the baseline. Moreover, the attention weights in the SSA are sparse. According to the dotted line in the figure, attention weights in the SSA are aligned with the ground truth. And most non-zero weights belong to the same class, even if the sound event overlaps with other classes of sound events or background sounds. This indicates that the sparse self-attention can model feature dependencies within a single event and between events of the same class more steadily, ignoring the irrelevant features. We can also find that the non-zero weights are higher than the baseline, which illustrates that global attention will weaken the attention weights of some important moments that need to be focused on. Therefore, embeddings generated by the sparse self-attention have more explicit semantics and stronger discrimination.

5. CONCLUSION

We attempted to improve the self-attention in the SS-SED model from a new perspective. To alleviate the impact of irrelevant features in the weighted summation process, we proposed a sparse self-attention, which uses the Sparsemax to make the model focus on features of the same class of sound events. Further, we proposed the Sparsemax with adjustable sparsity to solve the problem of excessive sparsity of the Sparsemax. Experimental results demonstrated the effectiveness of our method. In particular, our method has an advantage in dealing with short-duration sound events. Moreover, the distributions of attention weights showed that our approach could make it easier to distinguish overlapping events. In future work, we will select the optimal sparsity for each class of sound events to further improve the performance.

6. REFERENCES

[1] Y. Koizumi, Y. Kawaguchi, K. Imoto, et al., "Description and discussion on dcase2020 challenge task2: Unsupervised anomalous sound detection for machine condition monitoring," *arXiv preprint arXiv:2006.05822*, 2020.

[2] K. Ahmad and N. Conci, "How deep features have improved event recognition in multimedia: A survey," *ACM Trans. Multim. Comput. Commun. Appl.*, vol. 15, no. 2, pp. 39:1–39:27, 2019.

[3] J. P. Bello, C. Mydlarz, and J. Salamon, "Sound analysis in smart cities," in *Computational Analysis of Sound Scenes and Events*, pp. 373–397. 2018.

[4] M. Chavdar, B. Gerazov, Z. Ivanovski, and T. Kartalov, "Towards a system for automatic traffic sound event detection," in *28th Telecommunications Forum (TELFOR)*, 2020, pp. 1–4.

[5] G. Ciaburro, "Sound event detection in underground parking garage using convolutional neural network," *Big Data and Cognitive Computing*, vol. 4, no. 3, pp. 20, 2020.

[6] A. W. Ramadhan, A. Wijayanto, and H. Oktavianto, "Implementation of audio event recognition for the elderly home support using convolutional neural networks," in *International Electronics Symposium (IES)*, 2020, pp. 91–95.

[7] A. Mesaros, T. Heittola, A. Diment, B. Elizalde, A. Shah, E. Vincent, B. Raj, and T. Virtanen, "Dcase 2017 challenge setup: Tasks, datasets and baseline system," in *Proc. DCASE Workshop*, 2017.

[8] N. Turpault and R. Serizel, "Training sound event detection on a heterogeneous dataset," in *Proc. DCASE Workshop*, 2020.

[9] H. Sundar, M. Sun, and C. Wang, "Event specific attention for polyphonic sound event detection," in *Proc. Interspeech*, 2021, pp. 566–570.

[10] X. Zheng, Y. Song, I. McLoughlin, L. Liu, and L. Dai, "An improved mean teacher based method for large scale weakly labeled semi-supervised sound event detection," in *Proc. ICASSP*, 2021, pp. 356–360.

[11] L. Lin, X. Wang, H. Liu, and Y. Qian, "Guided learning for weakly-labeled semi-supervised sound event detection," in *Proc. ICASSP*, 2020, pp. 626–630.

[12] Q. Kong, Y. Xu, W. Wang, and M. D. Plumbley, "Sound event detection of weakly labelled data with cnn-transformer and automatic threshold optimization," *IEEE/ACM Trans. Audio Speech Lang. Process.*, vol. 28, pp. 2450–2460, 2020.

[13] K. Miyazaki, T. Komatsu, T. Hayashi, S. Watanabe, T. Toda, and K. Takeda, "Convolution-augmented transformer for semi-supervised sound event detection," Tech. Rep., DCASE, 2020.

[14] K. Miyazaki, T. Komatsu, T. Hayashi, S. Watanabe, T. Toda, and K. Takeda, "Weakly-supervised sound event detection with self-attention," in *Proc. ICASSP*, 2020, pp. 66–70.

[15] A. Pankajakshan, H. L. Bear, V. Subramanian, and E. Benetos, "Memory controlled sequential self attention for sound recognition," in *Proc. Interspeech*, 2020, pp. 831–835.

[16] A. F. T. Martins and R. F. Astudillo, "From softmax to sparsemax: A sparse model of attention and multi-label classification," in *Proc. ICML*, 2016, vol. 48, pp. 1614–1623.

[17] L. Delphin-Poulat and C. Plapous, "Mean teacher with data augmentation for dcase 2019 task 4," Tech. Rep., Orange Labs Lannion, France, June 2019.

[18] W. Wang and M. Á. Carreira-Perpiñán, "Projection onto the probability simplex: An efficient algorithm with a simple proof, and an application," *CoRR*, vol. abs/1309.1541, 2013.

[19] R. Serizel, N. Turpault, A. Shah, and J. Salamon, "Sound event detection in synthetic domestic environments," in *Proc. ICASSP*, 2020, pp. 86–90.

[20] H. Zhang, M. Cissé, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," in *Proc. ICLR*, 2018.

[21] A. Mesaros, T. Heittola, and T. Virtanen, "Metrics for polyphonic sound event detection," *Applied Sciences*, vol. 6, no. 6, pp. 162, 2016.

[22] Ç. Bilen, G. Ferroni, F. Tuveri, J. Azcarreta, and S. Krstulovic, "A framework for the robust evaluation of sound event detection," in *Proc. ICASSP*, 2020, pp. 61–65.

[23] C. Koh, Y. Chen, Y. Liu, and M. R. Bai, "Sound event detection by consistency training and pseudo-labeling with feature-pyramid convolutional recurrent neural networks," in *Proc. ICASSP*, 2021, pp. 376–380.