A Discriminative Convolutional Neural Network with Context-aware Attention

YUXIANG ZHOU, LEJIAN LIAO, YANG GAO, and HEYAN HUANG, Beijing Institute of Technology, China
XIAOCHI WEI, Baidu Inc.

Feature representation and feature extraction are two crucial procedures in text mining. Convolutional Neural Networks (CNN) have shown overwhelming success for text-mining tasks, since they are capable of efficiently extracting *n*-gram features from source data. However, vanilla CNN has its own weaknesses on feature representation and feature extraction. A certain amount of filters in CNN are inevitably duplicate and thus hinder to discriminatively represent a given text. In addition, most existing CNN models extract features in a fixed way (i.e., max pooling) that either limit the CNN to local optimum nor without considering the relation between all features, thereby unable to learn a contextual *n*-gram features adaptively. In this article, we propose a discriminative CNN with context-aware attention to solve the challenges of vanilla CNN. Specifically, our model mainly encourages discrimination across different filters via maximizing their earth mover distances and estimates the salience of feature candidates by considering the relation between context features. We validate carefully our findings against baselines on five benchmark datasets of classification and two datasets of summarization. The results of the experiments verify the competitive performance of our proposed model.

CCS Concepts: • **Information systems** → *Data mining*; *Document representation*;

Additional Key Words and Phrases: Text mining, convolution neural networks, attention method

ACM Reference format:
Yuxiang Zhou, Lejian Liao, Yang Gao, Heyan Huang, and Xiaochi Wei. 2020. A Discriminative Convolutional Neural Network with Context-aware Attention. *ACM Trans. Intell. Syst. Technol.* 11, 5, Article 57 (July 2020), 21 pages.
https://doi.org/10.1145/3397464

1 INTRODUCTION

Text-mining techniques are vital to support knowledge discovery, since diversity of text documents have been increased [1, 13, 32, 48]. Text mining has several important applications, like text classification [55, 59], summarization [2, 38], relation extraction [47, 63], and so on. In recent years,

This work was supported by the National Key Research and Development Program of China (Grant No. 2016YFB1000902), and the National Natural Science Foundation of China (Grant No. 61751201), and the Research Foundation of Beijing Municipal Science and Technology Commission (Grant No. Z181100008918002).
Authors' addresses: Y. Zhou, L. Liao, Y. Gao (corresponding author), and H. Huang, Beijing Institute of Technology, China; emails: {yxzhou, liaolj, gyang, hhy63}@bit.edu.cn; X. Wei, Baidu Inc.; email: weixiaochi@baidu.com.

2157-6904/2020/07-ART57 $15.00
https://doi.org/10.1145/3397464

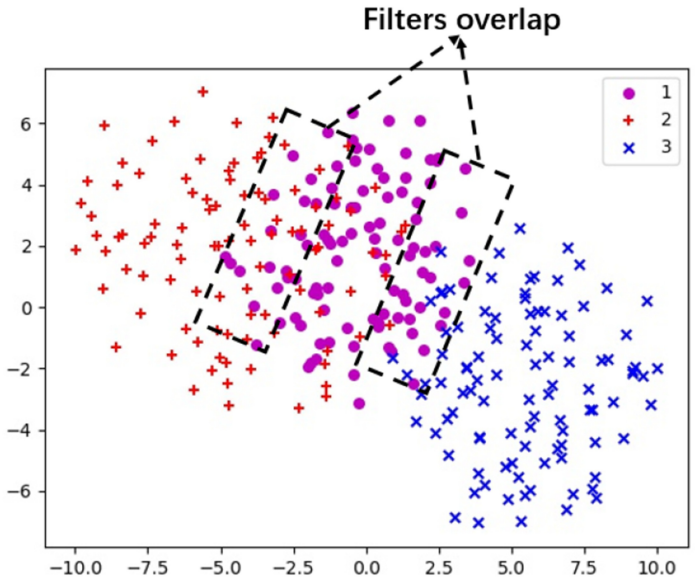


Fig. 1. t-SNE projections of the filters distributions for TREC dataset with MCCNN. Each color represents all the filters from one channel, i.e., purple color marks represent filters of size 1, red color marks represent filters of size 2, and blue color marks represent filters of size 3.

convolutional neural networks (CNN) have gained significant achievements in text mining, such as feature representation [24, 54, 60] and feature extraction [19, 55, 63]. They process input texts as representative *n*-grams and are flexible to extract salient features with respect to a specific task. But the vanilla CNN encounters the following two challenges.

Filter Distinction. As well known the aim of the convolutional operation is to find local patterns, such as the representative *n*-grams in source texts. It hence can be treated as feature representation, the quality of representation directly affects how does the system understand the semantics of texts. Redundant overlaps could be massively existed in the vanilla CNN model, revealed by filter distribution. As an example shown in Figure 1, we use t-SNE to project filter distributions of the vanilla for TREC dataset, the x-axis and y-axis represent the abscissa and ordinate of the filters in two-dimensional space. It can be found that filters of size 1, size 2 and size 3 are overlapped to each other, as a result, feature representations are inevitably non-discriminative after convoluted by these redundant filters. In addition, as the number of filters grows in a layer, some near-duplicates filters are evenly increases, as the conclusion drawn by [41]. The problem of a large amount of redundancy in the filters of CNN generally hinders to discriminatively represent a given text, thus degenerate the final performance of a specific task. As the example illustrated in Figure 2, vanilla CNN extracts reduplicated features such as *delighted*, *so delighted* and *is so delighted*. In this case, the information extracted by these three filters are quite redundant, and intuitively, the information extracted by the tri-gram filter covers the other two. However, there are some other useful *n*-gram features should also be extracted, such as *win* and *best actor* (e.g., see the second row of Figure 2). Several previous works have also demonstrated that too many redundancies among vanilla CNN filters will hinder the performance of itself [10, 31, 41], they target efficiency and attempt to increase the accuracy from the aspect of reducing the parameters of vanilla CNN, none of them have discussed the impacts of filter redundancy on textual tasks. In contrast, we add more parameters to avoid filter redundancy, a possible solution is to encourage

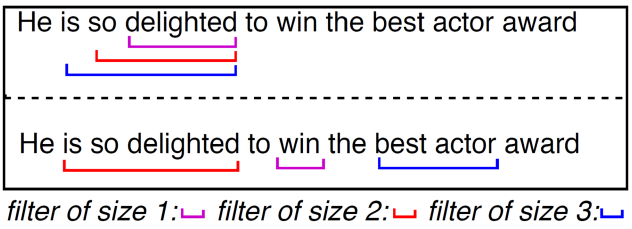


Fig. 2. An example sentence for illustrating the extracted *n*-gram features. First row shows the features extracted by regular CNN. Second row shows the expected features to be extracted.

divergence across different filters to discriminatively extract features from a sentence for task-specific use. More specifically, once the redundancy reduced, different filters will be far away from each other, then yielding a powerful text representation ability to distinguish the n-gram granularity of features, thereby, the classifier can identify this discriminative feature representation easily. From this point of view, our motivation is to solve the drawback of vanilla CNN in textual feature representation.

Feature Salience. Feature extraction is a crucial function of convolutional neural network. It refers to the procedure of generating a small set of new features by combining or transforming the original feature candidates. Pooling operation is usually viewed as a module of feature extraction in the CNN as it reduces the feature dimension from selecting the corresponding values from a local neighborhood. Max pooling and mean pooling are the two most widely applied pooling operations for computer vision tasks because of their simplicity and efficiency [43]. However, for text-specific tasks, both of them have their own drawbacks on extracting features from contextual related texts. Specifically, max pooling extracts the most representative information, since it only receives the highest feature value. Therefore, a great number of important features are dropped. While mean pooling extracts feature by averaging all information within the window, without considering their specific importance, which means noise has more of an impact and typically the features extracted are less salient. Some researchers [19, 58, 62] have explored different pooling operations as a way to extract features that are more relevant. For example, in the textual tasks [19] proposed a dynamic k-max pooling operator to capture varying sizes of features for sentiment classification. However, as it is a generalization of the max pooling operator, thus still suffers the shortcomings of traditional max pooling that have been clarified above. Consequently, a new feature extraction method that considers global context information and fine-grained calculation needed to be designed to upgrade vanilla CNN in text-mining task. We proposed a new pooling mechanism to comprehensively extract the salient features by considering all the relevant features in a global way.

Generally, the above challenges directly hinder the ability of CNN on feature representation and feature extraction. Recent works have demonstrated success with optimizing CNN from the two perspectives. For instance, transformable convolutions [59] adaptively reshape the filters to learn the varied patterns as the solution for representative features. Densely connected CNN [55] is equipped with an attention mechanism to extract multi-scale features. In spite of their success, few of them explicitly discuss the filter divergence or feature salience.

In this article, we devote to tackling with the aforementioned challenges of the vanilla CNN in terms of constructing **D**iscriminative filters and proposing a novel **C**ontex-aware **A**ttention, denoted as **DCA-CNN**. First, to eliminate the near-duplicate among filters, we design to construct distinguishable filters across different channels. Our idea is to proactively encourage filter divergence through interactively controlling the distances among components in the low-level of

convolved feature maps. As such, the proposed method yields diverse representations to distinguish the *n*-gram granularity of features in the texts. Second, to collect the overall context feature candidates globally and calculate their proper salience, we have designed a new context-aware attention mechanism that relates elements at different positions from a single sequence by computing the attention between each pair of feature candidates. In summary, the contributions of this article include:

- We analyze the filter redundancy in the vanilla CNN and propose to construct a module of discriminative filters that encourages dissimilarities across different filters to tackle the problem of semantic ambiguity.
- We propose a simple and effective context-aware attention mechanism that transforms each feature candidate to a salient one by considering all the relevant context features in a global way.
- For the experiments, we evaluate our model on two classical text-mining tasks, text classification, and summarization. Our experimental results on five text classification benchmark datasets and two summarization datasets examine the effectiveness of the proposed DCA-CNN model, and visualizations explicitly manifest the benefits of our model.

The rest of this article is organized as follows. Section 2 describes the related work about feature representation and feature extraction in text-mining applications. Section 3 defines the text classification problem and summarization problem and details our proposed DCA-CNN model. In Section 4, we present the experimental results and analysis, followed by a conclusion and future work in Section 5.

2 RELATED WORK

Feature representation is a vital component in text-mining applications. It has been extensively studied [45, 49, 50], a more effective and popular method being bag-of-words model [17], which represented text as the bag of its words, disregarding grammar and even word order but keeping multiplicity. Term frequency-inverse document frequency (tf-dif) [39], which intended to reflect how important a word was to a document in a collection or corpus, was another most commonly used method. Reference [41] investigated the presence of duplicate filters in neural networks in the computer vision task and observed that the number of duplicate filters increased proportionally with the number of filters in a layer, but they haven't discussed the impact of CNN redundancy on textual tasks. With the rapid development of deep neural networks [15], the CNN model has recently gained significant achievements on various text-mining tasks. Large number of researches have focused on improving the CNN from the aspect of augmenting feature representation. For example, Reference [20] applied multiple filters with different windows sizes to get multi-channel features for text classification. Reference [18] introduced novel continuous vector representations for semantically feature representations of sentences as a basis for measuring similarity, then employed the recursive auto-encoder to summarize documents. Reference [67] proposed a character-level CNN model that incorporated character-level features to CNN convolution layer. Reference [61] proposed a new neural architecture that constructed a document representation hierarchically to discover the relevant context information. Reference [24] adopted low-rank *n*-gram tensors instead of concatenating word representations to generate non-consecutive *n*-gram feature representation at the convolution stage. Reference [54] extended the CNN model with external knowledge to enrich features for short text classification. A new CNN paradigm [59] adaptively reshaped the filters to learn varied patterns as a solution for representative features, which captured the interaction inner filters and further fund flexible features rather than fixing *n*-grams. In addition, there are some recent techniques that learn discriminative feature representations, for

example, Reference [68] utilized an AL approach to learn independent word representation for discriminating the text class, but they only focused on learning word-level feature representation, which is insufficient to understand the full sentence. Thus, Reference [64] presented a general framework for short text classification by learning vector representations of both words and hidden topics together. From the perspective of the topic information, Reference [66] proposed a new topic-enhanced LSTM model to learn feature representation of documents. Reference [56] presented an approach to fine-grained recognition based on learning a discriminative mid-level feature representation within a CNN framework in an end-to-end fashion without extra annotation. Some works introduced regularizer to encourage the diversity among different feature spaces, Reference [26] took into account the inevitability regularization for general bidirectional sequence alignment models in machine translation task. Reference [27] introduced a disagreement regularization to explicitly encourage the diversity among multiple attention heads at different positions. In contrast to the mentioned approaches, we have explicitly discussed the issue of vanilla CNN redundancy in filtering or its influence on n-gram text representations.

Feature extraction is a vital preprocessing step for text-mining task used to solve the curse of dimensionality problem. In convolutional neural network, it refers to the course of generating a small set of new features by combining or transforming the original feature candidates [57]. Some works placed emphasis on extracting task-relevant features via different methods. Max pooling and mean pooling were two widely used methods because of their simplicity [3]. Later, Reference [46] designed an efficient feature extraction method called maximizing global information gain. Reference [19] used a dynamic k-max pooling operator to capture varying size of features for sentiment classification. A piecewise max pooling method [63] was designed to extract structural information for relation extraction. Reference [7] proposed a novel regularized CNN architectures to extract the spectral, spatial, and spectral-and-spatial-based deep features. Reference [5] presented a novel convolutional neural network structure that can extract more available features by unsupervised learning. Reference [65] introduced a new reinforcement-learning-based estimation network to extract the structured and meaningful features from raw text. Reference [6] proposed a new neural summarization model, which not just to pay attention to extract features of input documents with attention mechanism but also to distract the models to different features to better grasp the overall meaning of documents. However, the associated computational complexity increased as those models go deeper, which posed serious challenges in practical applications. Later, Reference [16] proposed a deep but low-complexity network architecture that converts discrete text to continuous representation by alternating a convolution block and a downsampling layer over and over. Reference [60] proposed a fast and effective CNN that applies gating mechanism to extract salient features of aspect for fine-grain sentiment analysis. Densely connected CNN [55] connected CNN by a hierarchical structure, which enables the model to select multi-scale features. Reference [8] proposed an effective method to extract salient features, their model trained by optimizing a new discriminative objective function that imposes a metric learning regularization term on the CNN features, apart from minimizing the classification error. It is essential to explore the effect of way on feature extraction. Therefore, in this article, we will target on augmenting the CNN from the aspects of text representation influenced by filters and an effective pooling mechanism for feature extraction.

3 PROPOSED MODEL

In this section, we first formalize the text classification and extractive summarization problem. After that, we detail our proposed discriminative convolutional neural network. At last, we describe the Context-aware Attention mechanism.

3.1 Problem Definition

Text classification, namely, text categorization, is defined as assigning predefined categories to sentences or text documents, where can be questions, technical reports, news stories, and so on, and categories are most often subjects or topics. Whatever the specific method employed, a text classification task starts with a training set $S = (s_1, \ldots, s_n)$ of sentences that are already labeled with a category $g \in G$. The task is to determine a classification model that is able to assign the correct class to a new sentence s of the domain.

The extractive summarization model reads the documents and selects a set of sentences to compose a summary. Existing models rely on feature representation and feature extraction methods to derive a meaningful representation of the documents, which is then used to label each sentence. Let $D = (d_1, \ldots, d_n)$ be a given document that consists of n sentences; the extractive summarization model outputs a binary decision of each sentence $Y = (y_1, \ldots, y_n)$, where n denotes the number of sentences in the document, and $y_i \in \{0, 1\}$ indicates whether sentence d_i is selected.

3.2 Model Architecture

The goal of this article is to incorporate the vanilla CNN with discriminative filters to eliminate the redundancy of convoluted filters and a novel context-aware attention to comprehensively extract salient features for a specific task, such as text classification, extractive summarization. The general architecture of the proposed DCA-CNN model is shown in Figure 3. The model begins with a tokenized sentence matrix whose rows indicate corresponding word embedding of each token. Then, it generates multi-channel representations by a convolution layer. Our proposed model constructs discriminative filters by maximizing the Earth Mover Distances (EMD) among the multi-channel features. Then Context-aware attention layer relates elements at different positions from a single sequence by computing the attention between each pair of feature candidates, thus collect the overall feature candidates globally and calculate their proper salience for the final feature by considering the relation between the feature candidates. Finally, the overall calculated feature salience is passed to a fully connected softmax layer to generate the final probability distribution over labels.

3.3 Convolutional Features Generation

We first let $\boldsymbol{x}_i \in \mathbb{R}^d$ be the d-dimensional word vector corresponding to the ith word in a sentence, and m denotes the number of words in the sentence. Then the sentence could be represented as a matrix $\boldsymbol{X} \in \mathbb{R}^{m \times d}$. Then a convolution operation with filter $\boldsymbol{W}_h \in \mathbb{R}^{h \times d}$ is applied to extract local features of word $\boldsymbol{X}_{i:i+h-1}$:

$$o_h^i = f(\boldsymbol{W}_h * \boldsymbol{X}[i : i + h - 1] + b_h), \tag{1}$$

where $\boldsymbol{W}_h$ and b_h are weight matrix of filters and the bias, f is a non-linear activation function such as tanh function $*$ performs the convolution operation. o_h^i is the ith feature convoluted by the filter of size h. This filter is applied to each window in the sentence to produce feature $\boldsymbol{c}_h = [o_h^1, o_h^2, \ldots, o_h^{m-h+1}]$. It is noted that multiple filters are usually used in convolutional layer of the CNN model to obtain multiple features. Therefore, the output feature map of channel h is matrix $\boldsymbol{C}_h \in \mathbb{R}^{(m-h+1) \times \ell}$. ℓ is the number of filters in each channel.

3.4 Discriminative Filters Construction

As we discussed in Section 1, redundancies of filters in CNN will lead the representation that learned by models to be non-discriminative, thus directly affecting the final performance of text classification. To eliminate the redundancies and maintain the distinctive part among filters, we design a discriminative filter module by maximizing a distance function as the objective. It is noted

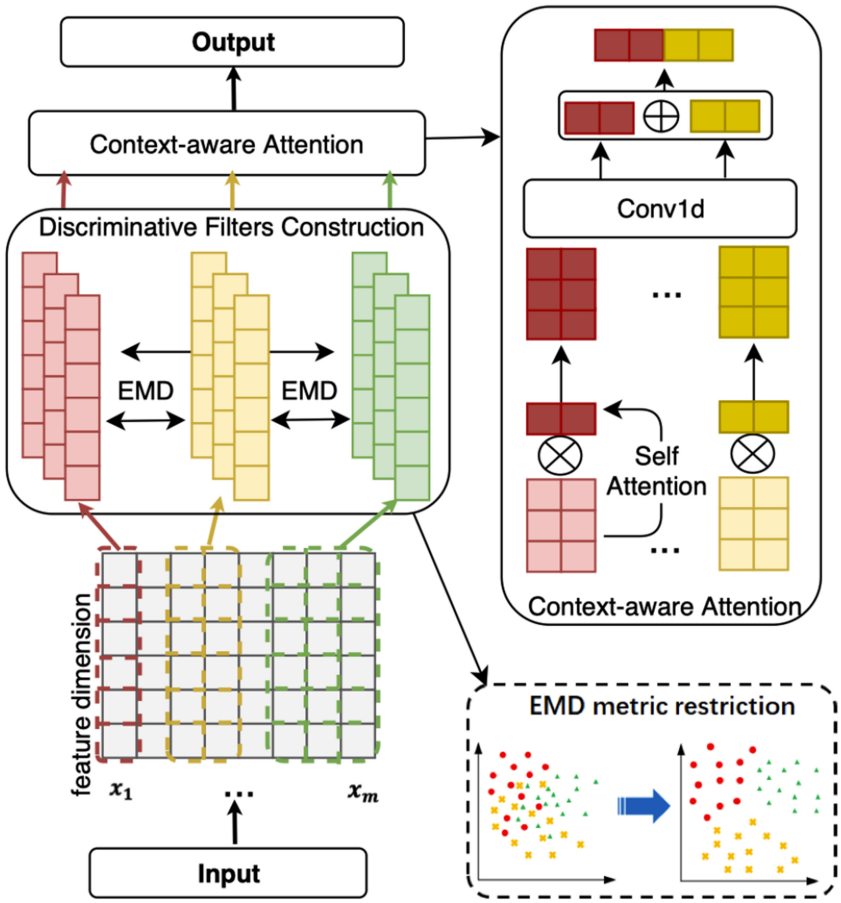


Fig. 3. Architecture of the proposed DCA-CNN model.

that we compute the distance by pairwise of feature maps instead of directly maximizing distance among filters. The reason is that the gradients generated from feature maps can be back propagated to the filter layer, since every component in the network is differentiable. Therefore, filter distribution is regulated to be separated.

Earth Mover Distance (EMD) is generally known as Wasserstein metric [25], it has been used to measure the distance between distributional word vectors [23] and the differences between LSTM hidden states [22]. As EMD is effective to measure the distance between hidden representation in NLP tasks, we adopt EMD as the distance function, which is designed to find the minimal cost to transform one distribution into the other. The distance metrics can be replaced by other distance metrics, such as Euclidean distance and cosine similarity. The reason why we use the EMD is explained by Figure 4. As the figure shows, the distance between $P(X)$ and $P(Y)$ is only $a + b$ computed by the EMD, since major components (in the middle) are quite similar of the two distributions. But the distance is relatively large by vector-based cosine similarity or Euclidean distance. Therefore, our model adopted by the EMD function can encourage the most maximized dispersion among the filters.

Given two different features $\boldsymbol{c}_h$ and $\boldsymbol{c}_k$, we define the distance between the two feature vectors (column of $\boldsymbol{C}$) that cross different channels as $d_{hk} = ||\boldsymbol{c}_h - \boldsymbol{c}_k||^2$. Let $\boldsymbol{T} \in \mathbb{R}^{\ell_h \times \ell_k}$ be a (sparse) flow

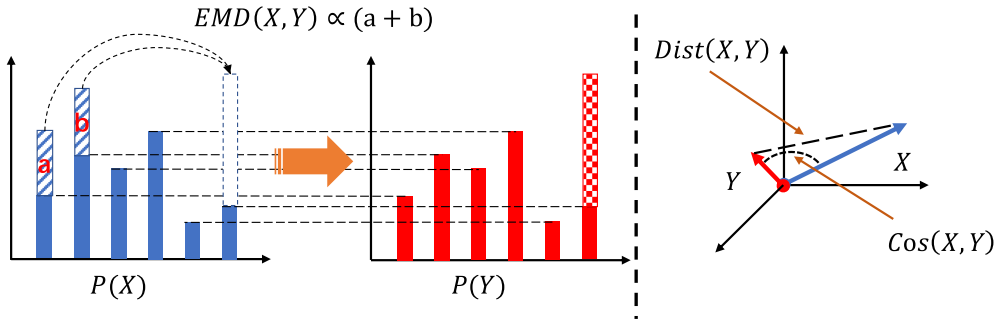


Fig. 4. Difference between EMD and vector-based cosine similarity and Euclidean distance.

matrix where $\boldsymbol{T}_{hk} \geq 0$ denotes how much effort the feature $\boldsymbol{c}_h$ travels to $\boldsymbol{c}_k$. Then, we define EMD as a minimization over an auxiliary transport matrix $\boldsymbol{T}$:

$$\boldsymbol{D}_{emd}(\boldsymbol{C}_h, \boldsymbol{C}_k) = \min_{\boldsymbol{T} \geq 0} \sum_{h=1}^{\ell_h} \sum_{k=1}^{\ell_k} \boldsymbol{T}_{hk}\, d_{hk}, \tag{2}$$

where ℓ_h and ℓ_k corresponding to the number of filters of size h and size k. $\sum_h T_{hk} = 1/\ell_h$ and $\sum_k T_{hk} = 1/\ell_k$ are constraints which normalize the total weight of two filters signatures (distribution) when calculating their earth mover distance to avoid favoring smaller signatures of the EMD transportation problem in Equation (2), they ensure the EMD a true metric, which is proved by Reference [42]. The optimization is special case of the earth mover's distance metric, a well-studied transportation problem for which specialized solvers have been developed [23].

3.4.1 Approximation. The time complexity of solving the EMD optimization via LP is $O(d^3 \log d)$, which is prohibitive, $d = max\{l_h, l_k\}$. We follow the approach of Reference [22] to alleviate the cubic time complexity EMD by adding an entropy regularization constraint to the EMD transportation problem:

$$\begin{aligned} \boldsymbol{D}'_{emd}(\boldsymbol{C}_h, \boldsymbol{C}_k) = &\min_{\boldsymbol{T} \geq 0} \sum_{n=1}^{\ell_h} \sum_{j=1}^{\ell_k} \boldsymbol{T}_{hk}\, d_{hk} \\ &+ \frac{1}{\lambda} \boldsymbol{T}_{hk} \log \boldsymbol{T}_{hk}, \end{aligned} \tag{3}$$

where $\lambda > 0$. Along with the λ increased, the relaxation is close to the original EMD. We follow the approximation taken by Reference [11]. To maximize the distance over the features, the objective is defined as

$$\mathcal{L}_{emd} = -\max \sum_{h,k=1,h\neq k}^{l} \boldsymbol{D}'_{emd}(\boldsymbol{C}_h, \boldsymbol{C}_k). \tag{4}$$

3.5 Context-aware Attention

In this section, we focus on elaborating the details of the proposed context-aware attention and how does it benefit to extract feature in our model. A key aspect of CNNs is the pooling layer, typically applied to extract features after the convolutional layers. In the pooling layer, the feature dimension is reduced by extracting the corresponding values from a local feature representation. Specifically, max pooling only extracts the highest feature value, mean pooling extracts features by averaging all information within the window. However, neither max pooling nor mean pooling calculates the dependencies between feature candidates with considering the contextual information,

which has proven useful for modeling dependencies among feature representations in text-mining tasks.

To tackle the aforementioned problems, we propose a novel context-aware attention mechanism that collects the overall feature candidates globally and calculates their proper salience for the final feature by considering the relation between feature candidates.

An attention function can be described as mapping a query and a set of key-value pairs to outputs, where the query, keys, values, and output are all vectors. The output is computed as a weighted sum of the values, where the weight assigned to each value is computed by a compatibility function of the query with the corresponding key. In practice, the attention function is usually computed on a set of queries simultaneously, packed together into a matrix $\boldsymbol{Q}$. The keys and values are also packed together into matrices $\boldsymbol{K}$ and $\boldsymbol{V}$. The matrix of outputs is defined as

$$Attention(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = \text{softmax}(\boldsymbol{Q}\boldsymbol{K}^T)\boldsymbol{V}. \tag{5}$$

Self-attention [53] is a special case of the attention mechanism introduced above. It replaces each vector $\boldsymbol{k} \in \boldsymbol{K}$ and $\boldsymbol{v} \in \boldsymbol{V}$ with the corresponding vector $\boldsymbol{q}$ from the source input itself matrix $\boldsymbol{Q}$. In our case, the matrix $\boldsymbol{Q}$, $\boldsymbol{K}$, and $\boldsymbol{V}$ are all come from the feature map $\boldsymbol{C}$.

Specifically, as shown is in Figure 3, in each context-attention block, a self-attention is used to calculate the relevance $\boldsymbol{\alpha_h}$ of feature map of channel h:

$$\boldsymbol{\alpha_h} = Attention(\boldsymbol{C}_h\boldsymbol{W}_h^Q, \boldsymbol{C}_h\boldsymbol{W}_h^K, \boldsymbol{C}_h\boldsymbol{W}_h^V), \tag{6}$$

where $\boldsymbol{W}_h^Q$, $\boldsymbol{W}_h^K$, and $\boldsymbol{W}_h^V$ are the weight metric. Then, the attention weight $\boldsymbol{\alpha_h}$ is used to adaptively re-weight the local n-gram features. As a result, the local feature candidates are updated by globally consider all the relevance at different positions from each row of feature map $\boldsymbol{C_h}$. A one-dimensional convolution is used to transform the updated $\boldsymbol{C_h}$ to final feature vector $\boldsymbol{z}_h$, defined as

$$\boldsymbol{z}_h = Conv1d(\boldsymbol{\alpha_h} \times \boldsymbol{o}_h). \tag{7}$$

Since we have the number of h feature vectors. A join layer then concatenates the feature vector from all channels as the final feature vector, denoted as

$$\boldsymbol{z} = \boldsymbol{z}_1 \oplus \boldsymbol{z}_2 \oplus \cdots \oplus \boldsymbol{z}_h, \tag{8}$$

where $\oplus$ refers to the operation that concatenates two vectors.

3.5.1 Softmax Layer. The final feature, $\boldsymbol{z}$, after applied global pooling operation, is fed into the classifier that is taken by a fully connected softmax layer. It computes the probability distribution over the labels:

$$p(y = j|\boldsymbol{z}) = \frac{\exp(\boldsymbol{z}, \boldsymbol{\theta}_j)}{\sum_{g=1}^{G} \exp(\boldsymbol{z}, \boldsymbol{\theta}_g)}, \tag{9}$$

where $\boldsymbol{\theta}_j$ is a weight vector of the jth class. G denotes the number of categories, where $j \in [1, G]$.

3.6 Training Model

Cross entropy is used as the loss function, which is represented as

$$\mathcal{L}_c = -\sum_{i=1}^{G} y_i \log(p_i), \tag{10}$$

where y_i is the ground truth, represented by one-hot vector, and p_i is the predicted probability for each class, computed as in Equation (9). The final objective function of our model can be defined as

$$\mathcal{L} = \arg\min_{\Theta} \left[\alpha\mathcal{L}_{emd} + (1 - \alpha)\mathcal{L}_c\right], \tag{11}$$

Table 1. Statistics of the Datasets for Text Classification

Datasets	Classes	Train/Test	Avg. words	Vocabulary
20NEWS	20	11,314/7,532	429	143,295
AGNEWS	4	120K/7.6K	45	67,746
TREC	6	5,952/500	10	9,524
MR	2	9,596/1,066*	24	18,767
SUBJ	2	9,000/1,000*	23	21,324

Train and Test denote the size of the train and the test set, respectively. "Avg. words" denotes the average sentence length. (*) means there was no standard train/test split and thus 10-fold cross validation was used.

where Θ is a set of all parameters in $\mathcal{L}_{emd}$ and $\mathcal{L}_c$, and α is a scale factor that balances the contribution of each component in the training process. To minimize objective in Equation (11), there are the following parameters in the model, denoted as

$$\Theta = \{\boldsymbol{W}_h, \boldsymbol{W}_h^Q, \boldsymbol{W}_h^K, \boldsymbol{W}_h^V, b_h, \boldsymbol{T}, \boldsymbol{\theta}_j, \boldsymbol{\theta}_g\}. \tag{12}$$

It is noteworthy that the parameter $\boldsymbol{T}$ is conducted by alternating optimization. First, the transport matrix $\boldsymbol{T}$ is optimized by fixing other parameters in Θ. Then, we maximize the distance between features by fixing $\boldsymbol{T}$ and propagate the loss back into the other parameters in Θ. The parameters of the network are optimized by Adam algorithm [21] with a mini-batch of 128 during the training process to optimize the parameters and the learning rate is 0.001. The training process lasts at most 20 epochs on all the datasets.

4 EXPERIMENTS AND RESULTS

In this section, we first evaluated the effectiveness of our methods in the text classification task, by comparing with their counterpart baselines. Moreover, to further investigate the ability to learn discriminative features with our model in other tasks, we tested our model's performance with an extractive summarization task. Then we analyzed the discrimination of filters across different channels and the advantages of applying context-aware attention in our model.

4.1 Datasets

For the text classification task, our proposed model was comparatively evaluated on various types of datasets, topic classification datasets, i.e., 20NEWS [23] and AGNEWS [67], a question classification dataset, i.e., TREC [28], and two semantic classification datasets, i.e., MR [36] and SUBJ [35]. Table 1 lists the statistics of the above datasets.

- **20NEWS:** A news categorization dataset. It has a collection of approximately 20,000 newsgroup documents, partitioned (nearly) evenly across 20 different newsgroups. Some of the newsgroups are close to each other, while others are highly unrelated. We use the *original* version of the 20 Newsgroup[1] dataset, which are grouped into 20 categories.
- **AGNEWS:** A large topic classification dataset. It consists of articles and descriptions of the AG's corpus of news,[2] topics of which include World, Sports, Business, and Sci/Tech.
- **TREC:** The task of TREC question dataset is to classify a question into 6 categories (i.e., person, location, numeric information).

[1] http://www.qwone.com/~jason/20Newsgroups/.
[2] http://www.di.unipi.it/~gulli/AG_corpus_of_news_articles.html.

Table 2. Statistics of the Datasets for Summarization Task

Datasets	Train	Vlidation	Test
CNN	90,266	1,220	1,093
DailyMail	196,961	12,148	10,397
CNN+DailyMail	287,227	13,368	11,490

Train, Validation, and Test denote the size of train, validation, and test set, respectively.

- **MR:** This dataset consists of movie reviews with one sentence per review. The classification task involves detecting positive/negative reviews.
- **SUBJ:** The task is to classify a sentence as being subjective or objective.

For the summarization task, we evaluated our models on the CNN and DailyMail news highlight datasets [14]. We used the standard splits for training, validation, and testing. We did not anonymize entities or lower case tokens. We followed previous studies [9, 34, 52] in assuming that the story highlights associated with each article are gold-standard abstractive summaries. During the training, we use these to generate high scoring extracts for them, but during testing, they are used as reference summaries to evaluate our models. Table 2 lists the statistics of the above datasets.

4.2 Comparative Methods

We choose two classic tasks in the text-mining area, such as text classification and extractive summarization. For text classification task, the baselines selected for comparison fall in two main categories: RNN methods and CNN methods. The RNN methods were:

- **Bi-LSTM:** A bi-directional Long-Short Term Memory (LSTM) [44] recurrent neural network model, which takes the whole text as a single sequence and output states of two hidden layers of opposite directions of all words.
- **Tree-LSTM:** A tree-structured LSTM model that relies on predefined parsing structure [51].
- **HS-LSTM:** A hierarchical structured LSTM that can build a structured representation by discovering hierarchical structures in sentence [65].
- **Self-Attentive:** A self-attention mechanism and a special regularization term are used to construct sentence embedding in an LSTM [30].

The CNN methods included:

- **MCCNN:** Multichannel CNN (MCCNN) [20] is a two-layer network, which represents texts by one convolution layer and one pooling layer for feature extraction. The features through a softmax function to generated the final classification.
- **TF-CNN:** A convolutional neural network model that replaces the convolution and pooling layers with the corresponding transformable modules [59].
- **DCCNN:** Dynamic convolutional neural network [19] is a deep-text classification model with seven layers, which is named after its dynamic k-max pooling.
- **Densely CNN:** A densely connected convolutional neural network with multi-scale feature attention for text classification [55].

For summarization task, we compared our model with the following extractive summarization baselines, since they rely heavily on feature representation and feature extraction:

- **LEAD3:** The commonly used baseline by only selecting the first three sentences as the summary.

Table 3. Classification Accuracy (%) Conducted by All the Models on Different Datasets

Model		SUBJ	MR	AGNEWS	TREC	20NEWS
RNN	Bi-LSTM	92.70	79.70	91.60	92.70	75.00
	Tree-LSTM [51]	93.20	80.70	91.80	-	-
	Self-Attentive LSTM [30]	92.50	80.10	91.10	-	-
	HS-LSTM [65]	93.70	82.10	92.50	-	-
CNN	MCCNN [20]	93.20	81.10	90.70	92.20	85.02
	DCCNN [19]	93.00	-	91.30	93.00	-
	TF-CNN [59]	95.00	-	-	93.50	-
	Densely CNN [55]	-	81.50	**93.60**	-	-
Ours	**DCA-CNN**	**95.00**	**82.20**	92.72	**93.80**	**87.00**

- **TEXTRANK:** An unsupervised method based on graphs proposed by Reference [33]. We implemented it by Gensim [40].
- **NN-SE:** [9] An extractive system that models document summarization as a sequence labeling task. We trained this baseline model with the same training data as our approach.
- **Distraction-M3:** A summarization system was proposed by Reference [6]. The system distracts the models to different content by attention control module to better grasp the overall meaning of input documents.
- **GBA-NN:** [52] A novel graph-based attention mechanism in a hierarchical encoder-decoder framework and proposed a hierarchical beam search algorithm to generate multi-sentence summary.
- **CNN+LSTM:** A hierarchical encoder-decoder extractive summarization system contained a vanilla CNN encoder module and an LSTM sentence extractor module. We trained this baseline model as same as Reference [34].
- **OURS+LSTM:** Our proposed model was composed of a DCA-CNN encoder and an LSTM decoder. The only difference between our model and the CNN+LSTM model was that we replaced the decoder by our proposed DCA-CNN.

4.3 Experiments Setting and Training Details

In our experiments, the hyper parameters of our model and all baseline methods were carefully tuned with the grid search to obtain the optimal performance. The word vectors were initialized by 300 dimensional Glove vectors [37]. The size of filter window (h) was chosen from 2 to 5 for TREC and from 1 to 7 for 20NEWS, AGNEWS, MR, and SUBJ. In each channel, the number of filters (l) were set to 100. We used $\alpha = 0.001$ for training the model. Considering the accuracy and complexity of EMD calculation, λ was set to 10 following previous work [22]. We employed dropout with a keep rate $p = 0.5$ on the penultimate layer. We trained all models for 20 epochs. For MR and SUBJ datasets, we replicated cross validation experiments 100 times, each replication was a 10-fold cross validation, wherein the folds were fixed, we reported average accuracy values observed. For other datasets, we selected the corresponding test set performance according to the best performance on the development set.

4.4 Performance in Text Classification Task

The overall performance in text classification tasks is summarized in Table 3. From the table, we observe that our model consistently outperforms the RNN-based models across all the datasets. Except for the Densely CNN [55] on AGNEWS dataset, our proposed model outperforms all the

Table 4. P-value Performance of our Model DCA-CNN Against the Vanilla CNN and the Next Best Baselines (i.e., TF-CNN and HS-LSTM) in Different Datasets

Model	SUBJ	MR	AGNEWS	TREC	20NEWS
MCCNN	1.73E-04	1.80E-02	2.43E-07	4.01E-04	1.87E-05
HS-LSTM	N	3.30E-02	3.00E-04	N	N
TF-CNN	1.82E-01	N	N	5.00E-03	N

Table cells are filled with N refer to not be the next best results.

Table 5. Classification Accuracy Conducted by BERT+CNN vs. BERT+DCA-CNN

Model	SUBJ	MR	AGNEWS	TREC	20NEWS
Glove+DCA-CNN	95.00	82.20	92.72	93.80	87.00
BERT+CNN	95.80	86.12	94.08	95.80	87.71
BERT+DCA-CNN	96.60	87.99	94.88	96.80	88.09

CNN-based baselines across all the datasets. Besides, we have performed a significance test on our model DCA-CNN with the vanilla CNN [20] and the next best baselines, respectively. The results are displayed in Table 4. We can find that with the exception of TF-CNN on the SUBJ dataset, all the p-values are substantially smaller than 0.05, indicating that the advantage of our model is statistically significant. Such solid results on such a variety of datasets demonstrate the competitive effectiveness of the proposed DCA-CNN on different datasets ranging from short-text classification to long-text classification and covering tasks, such as sentiment classification, topic and question classification.

Recently, pre-trained language models have benefited many NLP applications including text classification. It is unfair to directly compare with these pre-trained language model such as BERT [12], because they improve the performance by pre-training general word representations with large amount of data, and fine-tuning them according to the specific task. To be fair to the comparison, we use the output of the BERT[3] last layer as the input representation, followed by a vanilla CNN and our proposed DCA-CNN, respectively. The results are displayed in Table 5. From the table, we can find that BERT+DCA-CNN performed better than Glove+DCA-CNN on all dataset, which demonstrates using better representation such as BERT further boosts the performance of our proposed model. In addition, BERT+DCA-CNN outperformed the BERT+CNN across all datasets, which further demonstrates the effectiveness of our proposed model.

4.5 Performance in Summarization Task

For summarization task, we used Rouge scores [29] to evaluate the summarization performance, the overall performance is illustrated in Table 6. We observe that with the exception of the R1 value of GBA-NN [52] on the CNN dataset, our method has a considerable improvement over the other methods across all the datasets. An interesting observation is the results of DCA-CNN+LSTM are much higher than the CNN+LSTM model, which is similar to DCA-CNN performs much better than the vanilla CNN in text classification, which also demonstrates our proposed model is able to learn discriminative representation effectively.

[3]We use the base uncased BERT model implemented in PyTorch https://github.com/google-research/bert.

Table 6. Results on the CNN and DailyMail Test Sets

Models	CNN			DailyMail			CNN+DailyMail		
	R1	R2	RL	R1	R2	RL	R1	R2	RL
LEAD3[†]	-	-	-	-	-	-	39.20	15.70	35.50
TEXTRANK [33][†]	23.30	7.70	15.80	-	-	-	40.20	17.50	36.40
NN-SE [9][†]	28.40	10.00	25.00	36.20	15.20	32.90	35.50	14.70	32.20
Distraction-M3 [6][†]	27.10	8.20	18.70	-	-	-	-	-	-
GBA-NN [52]	**30.30**	9.80	20.00	-	-	-	38.10	13.90	34.00
CNN+LSTM [†]	27.00	9.80	24.20	37.90	14.40	33.70	37.00	13.70	33.40
DCA-CNN+LSTM	29.70	**10.70**	**26.20**	**40.00**	**16.10**	**36.10**	**40.50**	**17.80**	**37.10**

We report ROUGE-1 (R1), ROUGE-2 (R2), and ROUGE-L (RL) F1 scores. The marker - refer to results are not available. The mark [†] indicates that our proposed model is significantly better with $p < 0.05$ according to a one-tailed paired t-test.

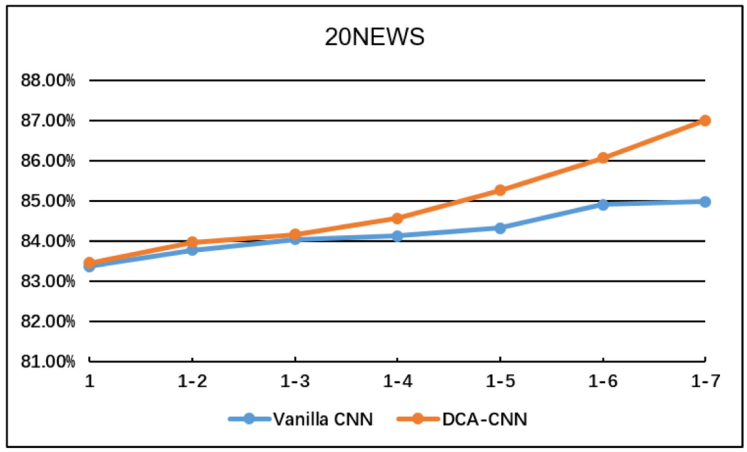
Table 7. Ablation Test: Applying Different Distance Metrics, i.e., Maximizing Euclidean Distance (D_E-CNN), Minimizing Cosine Similarity (D_C-CNN), Maximizing EMD Distance (D_{EMD}-CNN), and Different Pooling Strategies, i.e., Max Pooling, Mean Pooling, Context-aware Attention (CA)

Model	Strategy	SUBJ	MR	AGNEWS	TREC	20NEWS
CNN	Max pooling	93.40	81.50	90.40	93.60	84.20
	Mean pooling	90.70	77.20	90.11	90.60	83.10
	CA	93.80	80.60	92.10	93.80	86.30
D_E-CNN	Max pooling	93.60	80.11	91.40	92.30	86.21
	Mean pooling	93.00	79.92	90.97	91.40	86.15
	CA	93.60	80.86	91.60	93.20	86.84
D_C-CNN	Max pooling	93.80	80.11	92.10	92.60	86.27
	Mean pooling	93.20	80.10	92.10	92.00	85.35
	CA	93.70	80.95	93.60	93.00	86.94
D_{EMD}-CNN	Max pooling[†]	94.10	82.08	92.60	93.60	86.30
	Mean pooling[†]	93.70	81.14	92.59	93.20	86.20
	CA	**95.00**	**82.20**	**92.72**	**93.80**	**87.00**

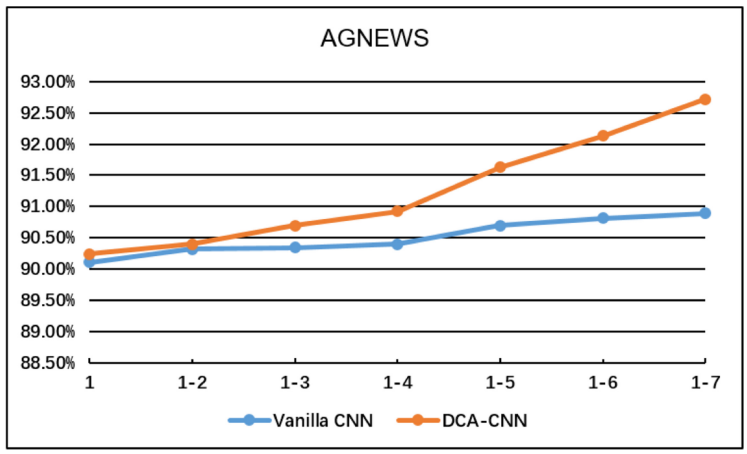
The best results are in bold. The mark [†] indicates that our proposed model is significantly better with $p < 0.05$ according to a one-tailed paired t-test.

4.6 Ablatioin Tests

4.6.1 Effect of Discriminative Filters. To validate the superiority of applying discriminative filters, we equip the vanilla CNN with discriminative filters that are conducted by different distance metrics, such as Euclidean distance (D_E-CNN), cosine similarity (D_C-CNN), and the EMD in our model (D_{EMD}-CNN). The performance is presented in Table 7. Overall, CNN models equipped with discriminative filters achieve better performance than the vanilla CNN over all the datasets. Besides, the D_{EMD}-CNN is the best of all the models equipped with the module of discriminative filters. The reasons are twofold, first, as we discuss in Section 3.4, it is useful to encourage large distance across the filters of the CNN. Second, distances computed by Euclidean and cosine similarity are better accurate for vector-based representations, while the EMD is designed for computing probability-based distance [4]. In the experiment, feature values after non-linear activation is always positive, thereby, are probability-based histograms. Therefore, it is verified that the module of discriminative filters is effective and the EMD is the best distance metric in our model. Moreover, we have tested to maximize filter distance inner each channel to eliminate



(a)



(b)

Fig. 5. Changes in performance between vanilla CNN and DCA-CNN at 20NEWS and AGNEWS as the number of different convolution kernels increased. The x-axis represents the size range of filters that we used.

the possibly inner duplicates. But the results show little improvement, and it brings a large computation cost. Therefore, it is not necessary to conduct inner channel divergence learning.

In addition, we investigated the changes in the performance of vanilla CNN and our proposed model at different datasets as the number of different convolution kernels increased. As shown in Figure 5, the results of vanilla CNN have a certain degree of improvement with the number of different convolution kernels increased. For example, in Figure 5(a), on the 20NEWS dataset, as the number of filters increased, the accuracy of vanilla CNN improved from below 84% to 85.02%. But in contrast, the accuracy of our proposed DCA-CNN significantly improved from below 84% to 87%. It demonstrates that redundant duplication massively exists in vanilla CNN when the filters increased, thus hinder the performance of the model. Our proposed model can learn discriminative filters, thus improve the quality of feature representation in text-mining tasks.

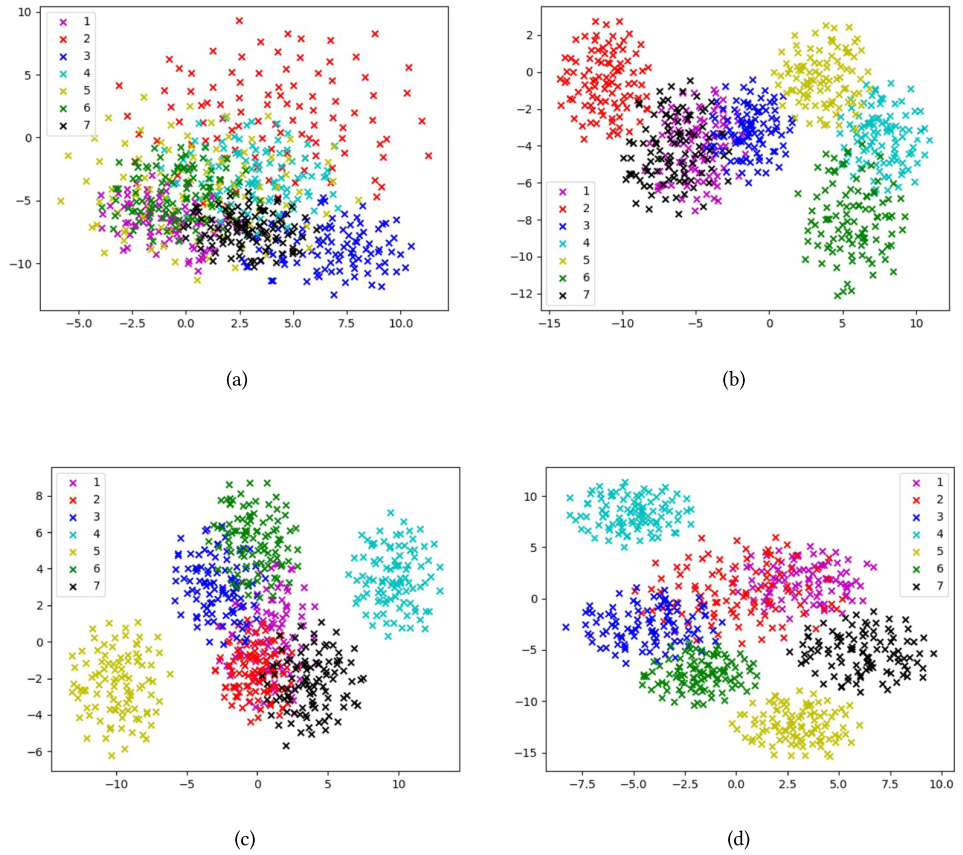


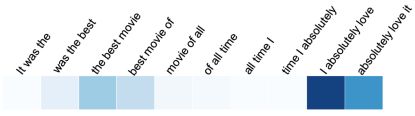
Fig. 6. Visualization of filter distribution via t-SNE projection. Each color represents all the filters from one channel, i.e., the purple × represents the filters that come from channel size = 1. The first two sub-figures, (a) and (b), are filter distributions before and after applying discriminative filter learning, respectively. Similar comparing group is conducted in sub-figures (c) and (d).

Visualization of Filter Divergence. In Figure 6, two groups of comparisons are conducted to visualize the difference between the vanilla CNN's filters and the learned discriminative filters by D_{EMD}-CNN. As demonstrated in Figures 6(a) and 6(c), redundant overlaps or duplication are massively existed in the vanilla CNN model, revealed by filter distribution. As a result, text representations are inevitably non-discriminative after convoluted by these redundant filters. But our D_{EMD}-CNN learns to construct quite separate filters contrast to the previous ones, shown in Figures 6(b) and 6(d), respectively. Therefore, it is worth believing that the representation convoluted by the filters of DCA-CNN can be more representative across different channels.

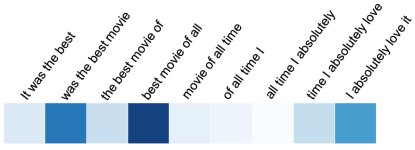
4.6.2 Effect of Context-aware Attention. To better investigate the effectiveness of the proposed context-aware attention mechanism, we equip vanilla CNN, D_E-CNN, D_C-CNN, and D_{EMD}-CNN with max pooling, mean pooling and context-aware attention (CA), respectively. We performed a comparison between these models on all the datasets, the comparison performance is presented in Table 7. It is observed that a model equipped with a Context-aware Attention (CA) always achieves the best performance across all datasets, except for the results of D_E-CNN and D_C-CNN on SUBJ dataset. The results verify that our proposed context-aware attention mechanism is more suitable to extract feature salience than the classical pooling strategies in most text classification cases.

(a) Attention weights of 1-gram features

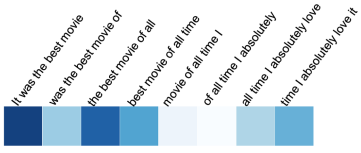
(b) Attention weights of 2-gram features



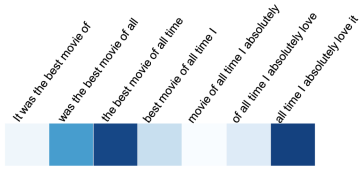
(c) Attention weights of 3-gram features



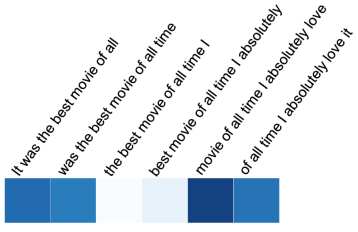
(d) Attention weights of 4-gram features



(e) Attention weights of 5-gram features



(f) Attention weights of 6-gram features



(g) Attention weights of 7-gram features

Fig. 7. Visualization of attention weights obtained by context-aware attention.

4.7 Case Study

To understand the effects of our proposed context-aware attention, we visualized the attention weights of the discriminative n-gram features by heatmaps. In particular, we focused on the relevance of α in Equation (6), it represents the dependency of each feature at the context level. We selected a sentence from the MR test set as an example. The sentence is *It was the best movie of all time, I absolutely loved it.* The relevance of the all the *n*-gram feature maps for this sentence is visualized by seaborn.[4] As shown in the Figure 7. The deeper color represents the more salient feature and vice-versa.

From Figure 7(a), we found that different 1-gram features, such as *best* and *loved*, that contain polarity are focused. The Figure 7(b) was shown that sentimental important 2-gram features such as *best movie* and *absolutely loved* get large attention weights, but irrelevant 2-gram features (*it was, was the, of all, etc.*) did not. As shown in the Figure 7(c), diverse 3-gram features, i.e., *I absolutely love, absolutely love it* got large attention from all other feature candidates. In addition, from

[4] https://seaborn.pydata.org/generated/seaborn.heatmap.html.

Table 8. Example Selected from CNN/DAILYMAIL to Visualize the Sentences Extracted by Our Model

Example 1	
Extracted sentences	–it's very vicar of dibley! it's **a wonderful act of** random **kindness** and i think that explains where it has come from. –residents of the picturesque Dorset village described the deed as **a wonderful act of** random **kindness**
Gold summary	local resident Tracey Feltham said: it's such **a wonderful act of kindness**
Example 2	
Extracted sentences	–the 27 year old left **will be out** for **at least four weeks after scans revealed** the severity of **the injury**. –the Paris Saint Germain defender is out for **at least four weeks after scans revealed** that the 27 year old suffered a torn hamstring in sunday's 3-2 win.
Gold summary	Luiz **will be out** for **at least four weeks after scans revealed the injury**.
Example 3	
Extracted sentences	–he will appear in court april 21 and plans to **plead guilty**. –he is due in court next week and plans to **plead guilty** to a misdemeanor.
Gold summary	has been charged with menacing he claims he will **plead guilty**.

Figures 7(d), 7(e), 7(f), and 7(g), we observed that reasonable 4-gram, 5-gram, 6-gram, and 7-gram feature candidates are easier to get high attention, respectively.

To deeply analyze our proposed model, we have added some case studies for summarization task. The selected examples are listed in Table 8. From the result, we observed that our proposed model can discriminatively extract the sentences that contain representative n-grams of various lengths in the gold summary. In example 1, the 1-gram feature *kindness* and the 4-gram feature *a wonderful act of* contained in each extracted sentence and those features are more likely to cover the gold summary. In example 2, the short 2-gram *the injury*, 3-gram *will be out*, and the long 7-gram *at least four weeks after scans revealed* are contained in the sentences selected by our model discriminatively. In example 3, the two sentences that contained an informative 2-gram feature *plead guilty* were extracted by our model as the candidate summaries.

5 CONCLUSION

In this article, we propose a novel DCA-CNN, which is equipped with a module of discriminative filters by maximizing the EMD distance of filters across different channels. A new context-aware attention mechanism is designed to comprehensively extract the salient features by considering all the relevant features in a global way. Leveraged by the two modules, our model is capable of capturing representative semantics and effectively computing feature salience for a specific task. Comparative experiments on classification and summarization tasks have demonstrated the promising performance, and visualizing analysis also explicitly manifested the benefits of our model. In the future, we will extend our method to adaptively extract flexible sizes of useful features with the context-aware mechanism. Furthermore, our model can be applied to other related tasks, such as relation classification, event extraction, and so on.

ACKNOWLEDGMENTS

The authors thank the anonymous reviewers for their insightful comments and valuable suggestions.

REFERENCES

[1] Charu C. Aggarwal. 2018. *Machine Learning for Text*. Springer.

[2] Ignacio Arroyo-Fernández, Arturo Curiel, and Carlos-Francisco Méndez-Cruz. 2019. Language features in extractive summarization: Humans vs. Machines. *Knowledge-Based Syst.* 180 (2019), 1–11.
[3] Y.-Lan Boureau, Jean Ponce, and Yann LeCun. 2010. A theoretical analysis of feature pooling in visual recognition. In *Proceedings of the 27th International Conference on Machine Learning (ICML'10)*. 111–118.
[4] Sung-Hyuk Cha and Sargur N. Srihari. 2002. On measuring the distance between histograms. *Pattern Recogn.* 35, 6 (2002), 1355–1370.
[5] Huan Chen, Licheng Jiao, Miaomiao Liang, Fang Liu, Shuyuan Yang, and Biao Hou. 2019. Fast unsupervised deep fusion network for change detection of multitemporal SAR images. *Neurocomputing* 332 (2019), 56–70.
[6] Qian Chen, Xiao-Dan Zhu, Zhen-Hua Ling, Si Wei, and Hui Jiang. 2016. Distraction-based neural networks for modeling document. In *Proceedings of the International Joint Conferences on Artificial Intelligence (IJCAI'16)*. 2754–2760.
[7] Yushi Chen, Hanlu Jiang, Chunyang Li, Xiuping Jia, and Pedram Ghamisi. 2016. Deep feature extraction and classification of hyperspectral images based on convolutional neural networks. *IEEE Trans. Geosci. Remote Sens.* 54, 10 (2016), 6232–6251.
[8] Gong Cheng, Ceyuan Yang, Xiwen Yao, Lei Guo, and Junwei Han. 2018. When deep learning meets metric learning: Remote sensing image scene classification via learning discriminative CNNs. *IEEE Trans. Geosci. Remote Sens.* 56, 5 (2018), 2811–2821.
[9] Jianpeng Cheng and Mirella Lapata. 2016. Neural summarization by extracting sentences and words. *Arxiv Preprint Arxiv:1603.07252*.
[10] Yu Cheng, Felix X. Yu, Rogerio S. Feris, Sanjiv Kumar, Alok Choudhary, and Shi-Fu Chang. 2015. An exploration of parameter redundancy in deep networks with circulant projections. In *Proceedings of the IEEE International Conference on Computer Vision*. 2857–2865.
[11] Marco Cuturi. 2013. Sinkhorn distances: Lightspeed computation of optimal transport. In *Advances in Neural Information Processing Systems*. MIT Press, 2292–2300.
[12] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *Arxiv Preprint Arxiv:1810.04805*.
[13] Ronen Feldman and James Sanger. 2007. *The Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data*. Cambridge University Press.
[14] Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Advances in Neural Information Processing Systems*. MIT Press, 1693–1701.
[15] Geoffrey E. Hinton and Ruslan R. Salakhutdinov. 2006. Reducing the dimensionality of data with neural networks. *Science* 313, 5786 (2006), 504–507.
[16] Rie Johnson and Tong Zhang. 2017. Deep pyramid convolutional neural networks for text categorization. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*. 562–570.
[17] Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2016. Bag of tricks for efficient text classification. *Arxiv Preprint Arxiv:1607.01759*.
[18] Mikael Kågebäck, Olof Mogren, Nina Tahmasebi, and Devdatt Dubhashi. 2014. Extractive summarization using continuous vector space models. In *Proceedings of the 2nd Workshop on Continuous Vector Space Models and their Compositionality (CVSC'14)*. 31–39.
[19] Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. 2014. A convolutional neural network for modelling sentences. *Arxiv Preprint Arxiv:1404.2188*.
[20] Yoon Kim. 2014. Convolutional neural networks for sentence classification. *Arxiv Preprint Arxiv:1408.5882*.
[21] Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *Arxiv Preprint Arxiv:1412.6980*.
[22] Sachin Kumar, Soumen Chakrabarti, and Shourya Roy. 2017. Earth mover's distance pooling over siamese LSTMs for automatic short answer grading. In *Proceedings of the International Joint Conferences on Artificial Intelligence (IJCAI'17)*. 2046–2052.
[23] Matt Kusner, Yu Sun, Nicholas Kolkin, and Kilian Weinberger. 2015. From word embeddings to document distances. In *Proceedings of the International Conference on Machine Learning*. 957–966.
[24] Tao Lei, Regina Barzilay, and Tommi Jaakkola. 2015. Molding cnns for text: Non-linear, non-consecutive convolutions. *Arxiv Preprint Arxiv:1508.04112*.
[25] Elizaveta Levina and Peter Bickel. 2001. The earth mover's distance is the mallows distance: Some insights from statistics. In *Proceedings Eighth IEEE International Conference on Computer Vision (ICCV'01)*, Vol. 2. IEEE, 251–256.
[26] Tomer Levinboim, Ashish Vaswani, and David Chiang. 2015. Model invertibility regularization: Sequence alignment with or without parallel data. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 609–618.
[27] Jian Li, Zhaopeng Tu, Baosong Yang, Michael R. Lyu, and Tong Zhang. 2018. Multi-head attention with disagreement regularization. *Arxiv Preprint Arxiv:1810.10183*.

[28] Xin Li and Dan Roth. 2002. Learning question classifiers. In *Proceedings of the 19th International Conference on Computational Linguistics*. Association for Computational Linguistics, 1–7.
[29] Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. *Text Summarization Branches Out* (2004), 74–81.
[30] Zhouhan Lin, Minwei Feng, Cicero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. 2017. A structured self-attentive sentence embedding. *Arxiv Preprint Arxiv:1703.03130*.
[31] Baoyuan Liu, Min Wang, Hassan Foroosh, Marshall Tappen, and Marianna Pensky. 2015. Sparse convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 806–814.
[32] Ling Luo, Xiang Ao, Feiyang Pan, Jin Wang, Tong Zhao, Ningzi Yu, and Qing He. 2018. Beyond polarity: Interpretable financial sentiment analysis with hierarchical query-driven attention. In *Proceedings of the International Joint Conferences on Artificial Intelligence (IJCAI'18)*. 4244–4250.
[33] Rada Mihalcea and Paul Tarau. 2004. Textrank: Bringing order into text. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.
[34] Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. Ranking sentences for extractive summarization with reinforcement learning. *Arxiv Preprint Arxiv:1802.08636*.
[35] Bo Pang and Lillian Lee. 2004. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, 271.
[36] Bo Pang and Lillian Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL'05)*. 115–124.
[37] Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP'14)*. 1532–1543.
[38] Ji-Peng Qiang, Ping Chen, Wei Ding, Fei Xie, and Xindong Wu. 2016. Multi-document summarization using closed patterns. *Knowledge-Based Syst.* 99 (2016), 28–38.
[39] Juan Ramos et al. 2003. Using tf-idf to determine word relevance in document queries. In *Proceedings of the 1st Instructional Conference on Machine Learning*, Vol. 242. Piscataway, NJ, 133–142.
[40] Radim Rehurek and Petr Sojka. 2010. Software framework for topic modelling with large corpora. In *Proceedings of the LREC Workshop on New Challenges for NLP Frameworks*. Citeseer.
[41] Aruni RoyChowdhury, Prakhar Sharma, Erik Learned-Miller, and Aruni Roy. 2017. Reducing duplicate filters in deep neural networks. In *Proceedings of the NIPS Workshop on Deep Learning: Bridging Theory and Practice*.
[42] Yossi Rubner, Carlo Tomasi, and Leonidas J. Guibas. 2000. The earth mover's distance as a metric for image retrieval. *Int. J. Comput. Vision* 40, 2 (2000), 99–121.
[43] Dominik Scherer, Andreas Müller, and Sven Behnke. 2010. Evaluation of pooling operations in convolutional architectures for object recognition. In *Proceedings of the International Conference on Artificial Neural Networks*. Springer, 92–101.
[44] Mike Schuster and Kuldip K. Paliwal. 1997. Bidirectional recurrent neural networks. *IEEE Trans. Signal Process.* 45, 11 (1997), 2673–2681.
[45] Ervin Sejdić, Igor Djurović, and Jin Jiang. 2009. Time-frequency feature representation using energy concentration: An overview of recent advances. *Digital Signal Process.* 19, 1 (2009), 153–183.
[46] Changxing Shang, Min Li, Shengzhong Feng, Qingshan Jiang, and Jianping Fan. 2013. Feature selection via maximizing global information gain for text classification. *Knowledge-Based Syst.* 54 (2013), 298–309.
[47] Ge Shi, Chong Feng, Lifu Huang, Boliang Zhang, Heng Ji, Lejian Liao, and Heyan Huang. 2018. Genre separation network with adversarial training for cross-genre relation extraction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. 1018–1023.
[48] Roberta A. Sinoara, Jose Camacho-Collados, Rafael G. Rossi, Roberto Navigli, and Solange O. Rezende. 2019. Knowledge-enhanced document embeddings for text classification. *Knowledge-Based Syst.* 163 (2019), 955–971.
[49] Heung-Il Suk, Seong-Whan Lee, Dinggang Shen, Alzheimer's Disease Neuroimaging Initiative, et al. 2014. Hierarchical feature representation and multimodal fusion with deep learning for AD/MCI diagnosis. *NeuroImage* 101 (2014), 569–582.
[50] Heung-Il Suk and Dinggang Shen. 2013. Deep learning-based feature representation for AD/MCI classification. In *Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 583–590.
[51] Kai Sheng Tai, Richard Socher, and Christopher D. Manning. 2015. Improved semantic representations from tree-structured long short-term memory networks. *Arxiv Preprint Arxiv:1503.00075*.
[52] Jiwei Tan, Xiaojun Wan, and Jianguo Xiao. 2017. Abstractive document summarization with a graph-based attentional neural model. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*. 1171–1181.

[53] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*. MIT Press, 5998–6008.

[54] Jin Wang, Zhongyuan Wang, Dawei Zhang, and Jun Yan. 2017. Combining knowledge with deep convolutional neural networks for short text classification. In *Proceedings of the International Joint Conferences on Artificial Intelligence (IJCAI'17)*, Vol. 350.

[55] Shiyao Wang, Minlie Huang, and Zhidong Deng. 2018. Densely connected CNN with multi-scale feature attention for text classification. In *Proceedings of the International Joint Conferences on Artificial Intelligence (IJCAI'18)*. 4468–4474.

[56] Yaming Wang, Vlad I. Morariu, and Larry S. Davis. 2018. Learning a discriminative filter bank within a CNN for fine-grained recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 4148–4157.

[57] Thomas Wiatowski and Helmut Bölcskei. 2017. A mathematical theory of deep convolutional neural networks for feature extraction. *IEEE Trans. Info. Theory* 64, 3 (2017), 1845–1866.

[58] Travis Williams and Robert Li. 2018. Wavelet pooling for convolutional neural networks. In *International Conference On Learning Representation*.

[59] Liqiang Xiao, Honglun Zhang, Wenqing Chen, Yongkun Wang, and Yaohui Jin. 2018. Transformable convolutional neural network for text classification. In *Proceedings of the International Joint Conferences on Artificial Intelligence (IJCAI'18)*. 4496–4502.

[60] Wei Xue and Tao Li. 2018. Aspect based sentiment analysis with gated convolutional networks. *Arxiv Preprint Arxiv:1805.07043*.

[61] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 1480–1489.

[62] Dingjun Yu, Hanli Wang, Peiqiu Chen, and Zhihua Wei. 2014. Mixed pooling for convolutional neural networks. In *Proceedings of the International Conference on Rough Sets and Knowledge Technology*. Springer, 364–375.

[63] Daojian Zeng, Kang Liu, Yubo Chen, and Jun Zhao. 2015. Distant supervision for relation extraction via piecewise convolutional neural networks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP'15)*. 1753–1762.

[64] Heng Zhang and Guoqiang Zhong. 2016. Improving short text classification by learning vector representations of both words and hidden topics. *Knowledge-Based Syst.* 102 (2016), 76–86.

[65] Tianyang Zhang, Minlie Huang, and Li Zhao. 2018. Learning structured representation for text classification via reinforcement learning. In *Proceedings of the Association for the Advancement of Artificial Intelligence (AAAI'18)*. AAAI.

[66] Wenyue Zhang, Yang Li, and Suge Wang. 2019. Learning document representation via topic-enhanced LSTM model. *Knowledge-Based Syst.* 174 (2019), 194–204.

[67] Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems*. MIT Press, 649–657.

[68] Ye Zhang, Matthew Lease, and Byron C. Wallace. 2017. Active discriminative text representation learning. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence*.

Received August 2019; revised March 2020; accepted April 2020