Information Sciences 548 (2021) 295–312

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Fusion of heterogeneous attention mechanisms in multi-view convolutional neural network for text classification

Yunji Liang [a,*], Huihui Li [a], Bin Guo [a], Zhiwen Yu [a], Xiaolong Zheng [b,d,*], Sagar Samtani [c], Daniel D. Zeng [b,d]

[a] School of Computer Science, Northwestern Polytechnical Univeristy, Xi'an, ShaanXi, China
[b] State Key Laboratory of Management and Control for Complex Systems, Institute of Automation Chinese Academy of Sciences, Beijing, China
[c] Operations and Decision Technologies Department, Kelley School of Business, Indiana University, USA
[d] University of Chinese Academy of Sciences, Beijing, China

ARTICLE INFO

Article history:
Received 7 March 2020
Received in revised form 10 October 2020
Accepted 12 October 2020
Available online 17 October 2020

Keywords:
View attention
Spatial attention
Multi-view representation
Series and parallel connection
Conventional neural network
Text classification

ABSTRACT

The rapid proliferation of user generated content has given rise to large volumes of text corpora. Increasingly, scholars, researchers, and organizations employ text classification to mine novel insights for high-impact applications. Despite their prevalence, conventional text classification methods rely on labor-intensive feature engineering efforts that are task specific, omit long-term relationships, and are not suitable for the rapidly evolving domains. While an increasing body of deep learning and attention mechanism literature aim to address these issues, extant methods often represent text as a single view and omit multiple sets of features at varying levels of granularity. Recognizing that these issues often result in performance degradations, we propose a novel Spatial View Attention Convolutional Neural Network (SVA-CNN). SVA-CNN leverages an innovative and carefully designed set of multi-view representation learning, a combination of heterogeneous attention mechanisms and CNN-based operations to automatically extract and weight multiple granularities and fine-grained representations. Rigorously evaluating SVA-CNN against prevailing text classification methods on five large-scale benchmark datasets indicates its ability to outperform extant deep learning-based classification methods in both performance and training time for document classification, sentiment analysis, and thematic identification applications. To facilitate model reproducibility and extensions, SVA-CNN's source code is also available via GitHub.

1. Introduction

The rapid proliferation of user generated content has given rise to large volumes of text corpora. Increasingly, researchers, scholars, and organizations are employing natural language processing (NLP) to mine novel insights from news sources, social media, and other forms of computer mediated communication. A prevailing NLP task is machine learning-based text classification. To date, text classification has been successfully applied in many high-impact applications including sentiment

* Corresponding authors at: State Key Laboratory of Management and Control for Complex Systems, Institute of Automation Chinese Academy of Sciences, Beijing, China (X. Zheng).
E-mail addresses: liangyunji@nwpu.edu.cn (Y. Liang), xiaolong.zheng@ia.ac.cn (X. Zheng).

https://doi.org/10.1016/j.ins.2020.10.021
0020-0255/

analysis [1,2], illegal sales detection [3,4], early screening of mental health issues [5,6], fake news detection [7,8], deviant content identification [9,10], and numerous others.

Fundamentally, text classification aims to learn a function that assigns a class label to a text instance based on the features of the instance. Conventional approaches handcraft multiple categories of features (e.g., lexical, syntactic, term frequencies, etc.) and feed them into machine learning algorithms such as support vector machine (SVM), Naïve Bayes, random forest, decision tree, and others [11]. However, hand-crafting feature extraction is a labor-intensive task that often requires significant domain expertise. Moreover, selected feature sets are often task specific, omit long-term relationships within text corpora, and are often brittle and ever-changing in the rapidly evolving applications.

Recognizing these challenges, scholars have increasingly turning to deep learning-based techniques such as Long Short-Term Memory (LSTM), Recurrent Neural Network (RNN), Gated Recurrent Unit (GRU), and Convolutional Neural Network (CNN) [1,12–14]. Such methods use non-linear activation functions, back-propagation, and error correction computations to automatically learn salient feature representations from a given text input. Recent years have augmented these models by incorporating attention mechanisms that compute a score for each token within an input vector based on how important the token for the final decision.

Despite their advantages, extant deep learning-based approaches suffer from several key drawbacks. First, input text is often represented using a single view (i.e., single set of features), rather than comprehensively examining multiple different levels of granularity (e.g., *n*-grams, phrases, etc.). Fine-grained text representations can enhance overall text classification performance. Second, although numerous attention mechanisms are proposed to learn the weights of tokens, they either quantify the significant of words or measure the association between tokens and the specific tasks in one single view. How to construct attention mechanisms that capture the latent interaction among multiple views is unclear. Finally, and relatedly, how to systematically combine multi-view text representation learning and attention mechanism in a manner that maximizes predictive power while minimizing training time is unknown.

Motivated by the aforementioned limitations, we propose a novel Spatial View Attention Convolutional Neural Network (SVA-CNN) framework. SVA-CNN leverages a carefully designed set of multi-view representation learning, a heterogeneous combination of novel attention mechanisms, and CNN-based operations. Through the design and rigorous evaluations of SVA-CNN, this paper makes several key contributions to text classification and interpretable deep learning-based methodologies. These include:

- A novel multi-view representation of text to automatically extract multiple granularities and fine-grained representations (e.g., phrases, *n*-grams, etc.) from a given text input without manual feature engineering;
- An innovative spatial attention mechanism that automatically learns the importance of words and phrases from the multi-view representation;
- A thorough comparison of SVA-CNN's predictive capability against prevailing text classification methods on five large-scale benchmark datasets for document classification, sentiment analysis, and thematic identification applications;
- A systematic investigation into the effect of series-parallel connection strategies of spatial and view attention mechanism on overall model performance and training time;
- A public release of the SVA-CNN model via the popular social coding repository, GitHub, to facilitate future model reproducibility and extensions.

The remainder of this paper is organized as follows. In Section 2, we ground the proposed SVA-CNN model by formally reviewing related work in text classification, multi-view representation learning, and attention mechanisms. Section 3 summarizes key research gaps from extant literature, and presents a systematic overview of the proposed SVA-CNN. Section 4 delves into the core novelties of the proposed model by formally denoting its key operations. Section 5 summarizes the overall evaluation set up, benchmark datasets, and key results. Section 6 discusses the implications of these results and illustrates the potential practical utility of the proposed model. Finally, Section 7 offers concluding remarks and summarizes promising future directions for research.

2. Related work

2.1. Text classification

Formally, text classification can be formulated as follows. Given a corpus $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$, where $\mathbf{x}_i = [x_i^1, \ldots, x_i^m]$ is the m-dimensional feature vector of text; $y_i \in \mathbb{C}$ is the corresponding label of given textual data; $\mathbb{C}$ refers to the set of classes. The goal of one classification problem is to learn a mapping function that predicts the label for one given textual sequence with least biases. Typically, text classification algorithms can be categorized into two major groups (conventional and deep learning-based solutions). Both approaches have been applied extensively in numerous application areas. We summarize the key principles of each approach in the following sub-sections.

2.1.1. Conventional text classification

Numerous hand-crafted features such as word frequency, grammatical, lexical, psychological, and structural features are proposed. Prevailing supervised classification algorithms include SVM [15], random forest [11], Naïve Bayes [11], and ensemble learning [16] are employed to bridge the mapping between feature sets and labels. However, text classification based on conventional classification algorithms rely on feature engineering, resulting in algorithms designed to detect specific indicators. Hand-crafting feature is time consuming, labor intensive, and not suitable for rapidly evolving domains. Due to the diversity, ambiguousness of words and dynamics of languages, to pre-define one complete set of features for large-scale textual corpus is infeasible. On the other hand, supervised learning algorithms are based on samples made from existing corpora. Thus, those approaches are useful to detect the content similar to existing corpora, and are less intelligent in detecting emerging instances.

2.1.2. Deep learning-based text classification

RNN and its variants including LSTM and GRU are the prevailing models for sentiment analysis [1], translation [17] and sequence tagging [18,13] follows. However, variants of RNN-based models are subject to the loss of long-distance contextual information and the focus of input vectors is equally scattered. CNN-based models are widely adopted for text classification [19–21], relation extraction and classification [22,23]. CNNs use a series of padding, convolutional, and pooling layers to automatically learn features from raw data (e.g., input text) in a matrix form. For example, Li et al. proposed a CNN-based model with two cascading convolutional layers to learn the word-phrase relation and the phrase-sentence relation for social emotion classification [20]. CNN-based solutions are often preferred over RNN-based models in situations where training time is an essential consideration [24]. However, the CNN-based solutions are often used in a single-view fashion, thus potentially omitting critical features (e.g., long term dependencies) within the input text. Meanwhile, existing studies primarily use concatenation to fuse heterogeneous features. How to efficient combine heterogeneous features will be helpful for improving the performance. An increasingly viable remedy for this issue is multi-view representation learning, which we review next.

2.2. Multi-view representation learning

Multi-view representation learning is an emerging topic in machine learning that fuses multiple views (feature sets) to improve the performance. In contrast to single view learning, multi-view learning introduces a function to each view and jointly optimizes all the functions to exploit the redundant views of the same input data and improve the learning performance. Multi-view representation learning can be categorized into two groups: multi-modal methods and multi-view methods.

2.2.1. Multi-modal methods

Multi-modality indicates one signal described by different medias. For example, happiness can be expressed by facial expression, acoustic tones, and linguistic clues. Numerous studies show that the significant performance improvements have been observed by fusing the disparate, yet related feature sets expressed through different medias [25,26]. For example, the fusion of facial emotion and vocal features was studied for the detection of depression disorder [27]. Hassanpour et al. fused the images and texts posted on Instagram, and utilized the jointly deep learning model for the large-scale screening of substance abuse including drug abuse and alcohol addiction [28]. Chancellor et al. proposed a multi-modal classifier to jointly learn both textual and visual characteristics from Tumblr to automatically distinguish the pro-eating disorder posts [29]. Jiang et al. proposed the latent topic text representation learning to provide an effective text representation and text measurement with latent topics for text classification task [30].

2.2.2. Multi-view methods

Multiple views refers to one signal to be described at different granularities. Multi-view representation learning is widely adopted in computer vision tasks. Liu et al. proposed an example-based multi-view domain generalization framework for visual recognition by learning robust classifier that can be generalized to arbitrary target domain based on the training samples with multiple types of features [31]. Zhang et al. proposed a deep neural network (DNN)-driven feature learning method for multi-view facial expression recognition [32].

In NLP, Qiu et al. proposed a multi-view annotation framework for Chinese treebanking, which integrates the dependency structure and phrase structure [33]. Dhillon et al. introduced a context-specific word embedding which measures the interaction among views by the canonical correlation analysis to find the latent structure [34]. The proposed multi-view word embedding performed well on named entity recognition and chunking tasks. However, for text classification, the sequential input data is converted into sequential word-level vectors, which only provide a single view on the word level. These approaches overlook the latent interaction among complex semantic groups. How to construct and apply the multi-view representation for text classification tasks is an open issue.

2.3. Attention mechanisms in NLP

Over the last few years, attention mechanisms have found broad applications in numerous NLP tasks. Attention mechanisms aim to learn to what extent words are associated with the specific task [35]. The basic idea is to compute an attention score for each token, and to modulate the input vectors accordingly. To date, numerous attention mechanisms have been proposed to preserve the context dependency [36], to capture both the global and local features in image caption tasks [37], and to learn the hierarchial fine-grained features for document classification [38]. Various attention mechanisms were proposed to encode the sequence data based on the weight each element is assigned. Among them, Bahdanau et al. introduced the attention mechanism in neural machine translation to build explicit word alignment during decoding [39]. To preserve the long-distance context of tokens, Wu et al. proposed the context attention to improve the word representation generated by word2vec [36]. For document classification, Yang et al. introduced the hierarchical attention to construct the bottom-up representations ranging from word-level to sentence-level vectors [38]. Lately, the self-attention has been introduced to learn a distributed relation representation with respect to all other tokens, and has achieved the state-of-the-art performance in translation [40].

However, the prior attention mechanisms either quantify the significant of words or measure the association between tokens and the specific tasks in one single view. As a result, these approaches do not fit well for the multi-view data for several reasons. First, the prior studies for multi-view data consider concatenating all multiple views into one single view and applies single-view learning algorithms directly [28,29]. However, the disadvantages of this method include the over-fitting problem on comparatively small training sets and the missing of specific statistical property of each view [25]. Second, most existing attention-based models only take into account the semantic features by modulating the sentence context into the last convolutional layer feature map via spatially attentive weights [28,29]. While multiple attention mechanisms including spatial attention, semantic attention, and multi-layer attention perform well to preserve the fine-grained features from different views, how to connect heterogeneous attention mechanisms for multi-view data has not been studied yet.

3. Overview of the proposed SVA-CNN model

To date, most studies only focus on single view data of text, and design numerous attention mechanisms to learn the weights of word pairs. **However, how to construct the multi-view text representation for fine-grained feature extraction and preserve the latent interaction among different views are rarely studies**. Therefore, we are interested in studying the multi-view representation and heterogeneous attention mechanisms for text classification. Specifically, we explore the multi-view representation of textual data, propose the context attention, spatial attention, and view attention mechanisms to preserve the complex latent interaction among different-granularity semantic groups, and study the connection of heterogeneous attention mechanisms to find the optimized connection for multi-view text classification.

In particular, we propose a novel SVA-CNN deep learning architecture. SVA-CNN leverages a multi-view representation of text to learn high-level features. The multi-view representation aims to construct the collection of vectorized n-gram features that enrich the feature representation in different granularities. Based on the multi-view representation, the original text can be formulated in word level, phrase level, and n consecutive words, which make it possible to learn the semantics from different granularities. Spatial attention and view attention mechanisms are proposed to preserve the latent interaction among different-granularity semantic groups. Specifically, the spatial attention aims to learn the long-term dependencies among n-gram features by quantifying the importance of n-gram features for text classification. The view attention mechanism characterizes which channel of the n-gram features are important. The view attention mechanism aims to learn the multi-granularity semantic groups for text classification. Finally, the series-parallel connection of heterogeneous attention mechanisms aims to find an efficient way to fuse attention mechanisms with multi-view text classification.

As shown in Fig. 1, the workflow is controlled by the two switches (*Switch A* and *Switch B*). The context attention aims to preserve the long-term dependency between words by scaling the weights of surrounding tokens. Whether the context attention is adopted to modulate word vectors or not is controlled by *Switch A*. We introduce the attention pool to provide a variety of attention mechanisms for the enhanced performance. In the attention pool, we introduce two basic attention mechanisms: *spatial attention* to learn the importance of semantic features and *view attention* to learn the significance of views from multi-view data. Apart from the spatial attention and view attention, we study the combination of spatial and view attention mechanisms in different ways including series connection and parallel connection. The series connection can be formulated as $\mathbf{V}^{(0)} \rightarrow \mathbf{V}^{(1)} = f_s^{se}\left(\mathbf{V}^{(0)}\right) \rightarrow f_v^{se}\left(\mathbf{V}^{(1)}\right)$, where $\mathbf{V}^{(i)}$ denotes the feature map of multi-view data on the i-th layer; $f(\cdot)$ symbolizes a set of attention mechanisms; The subscript of $f(\cdot)$ refers to attention mechanisms (s for spatial attention, v for view attention); The superscript of $f(\cdot)$ refers to how attention mechanisms are connected (se for series connection, and pa for parallel connection). Similarly, we can organize the attention mechanisms in another way by $\mathbf{V}^{(0)} \rightarrow \mathbf{V}^{(1)} = f_v^{se}\left(\mathbf{V}^{(0)}\right) \rightarrow f_s^{se}\left(\mathbf{V}^{(1)}\right)$. For parallel-connected attention, the spatial attention and view attention take the feature map $\mathbf{V}^{(0)}$ as input respectively and conduct specified transformation in a parallel manner. The parallel connection is formulated as $\mathbf{V} \begin{matrix} \nearrow f_s^{pa}\left(\mathbf{V}^{(0)}\right) \searrow \\ \searrow f_v^{pa}\left(\mathbf{V}^{(0)}\right) \nearrow \end{matrix} \mathrm{F}\left(f_s^{pa}\left(\mathbf{V}^{(0)}\right), f_v^{pa}\left(\mathbf{V}^{(0)}\right)\right)$. To evaluate the effects of heterogeneous attention mechanisms, we use *Switch B* to control which one is chosen from the attention pool.

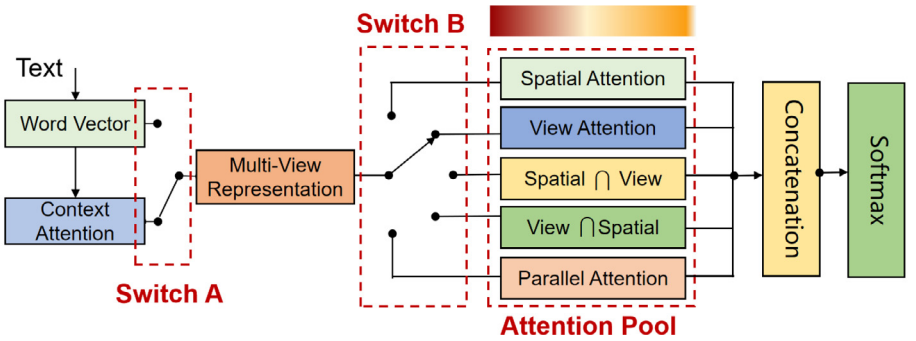

Fig. 1. Overview of the proposed SVA-CNN. Switch A is used to control whether the context attention is applied to modulate the word vector; Switch B aims to actuate the transition among different attention mechanisms in the attention pool by flipping the switch to different positions.

As the newly constructed feature map is represented in multiple views, the concatenation layer fuses the multi-view feature maps to build one comprehensive feature map for text classification. The dropout layer is included to avoid the overfitting problem. The softmax layer is provided to find the maximum likelihood estimation of classes. The details of primary components in Fig. 1 are elaborated in the following sections. All notations used in this paper are summarized in Table 1.

4. Heterogeneous attention mechanisms

4.1. Context attention

Conventional word vectors only focus on the local structural features based on the co-occurrence of words in the context windows. However, they often overlook the long-term dependencies between non-consecutive words. To address this problem, Zhao and Wu proposed the context attention mechanism to capture the long-term correlation among words [36]. In this paper, we apply the context attention mechanism to learn the fine-grained word representation. The key idea of context attention is described below.

For a given textual input, the text matrix is denoted as $\mathbf{A} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_i, \ldots, \mathbf{x}_l]^{\mathbf{T}}$, where $\mathbf{x}_i \in \mathbb{R}^d$ refers to the d-dimensional word vector corresponding to the i-th token in the text sequence; l is the total number of tokens in the given sequence. Given the i-th token vectorized as $\mathbf{x}_i$, the context attention determines which contextual vectors $\mathbf{x}_{j,j\neq i}, 1 \leqslant j \leqslant l$ should be paid more attention. The context vector $\widehat{\mathbf{x}}_i$ is formulated as $\widehat{\mathbf{x}}_i = \sum_{j\neq i} \alpha_{i,j} \cdot \mathbf{x}_j$, where $\alpha_{i,j} \geqslant 0, 1 \leqslant j \leqslant l$ are the attention weights, and $\sum_j \alpha_{i,j} = 1$. The value of $\alpha_{i,j}$ can be quantified by the softmax layer as shown in Eq. (1), where $score(\cdot)$ is used to quantify the correlation of word pairs $(\mathbf{x}_i, \mathbf{x}_j)$. The definition of $score(\cdot)$ is shown in Eq. (2), where $\oplus$ refers to the concatenation.

$$\alpha_{i,j} = \frac{exp\left(score\left(\mathbf{x}_i, \mathbf{x}_j\right)\right)}{\sum_{j'=1, j'\neq i}^{l} exp\left(score\left(\mathbf{x}_i, \mathbf{x}_{j'}\right)\right)} \tag{1}$$

$$score\left(\mathbf{x}_i, \mathbf{x}_j\right) = tanh\left(\mathbf{W}^{\alpha}\left[\mathbf{x}_i \oplus \mathbf{x}_j\right]\right) \tag{2}$$

Based on Eqs. (1) and (2), the context vector $\widehat{\mathbf{x}}_i \in \mathbb{R}^{d_s}$ of word vector $\mathbf{x}_i$ is learned based on the context attention mechanism, where d_s is the width of context vector. To preserve both the local structural features and long-term dependency of text matrix $\mathbf{A}$, we concatenate $\widehat{\mathbf{x}}_i$ and $\mathbf{x}_i$ as the newly generated representation of words $\mathbf{x}'_i = \mathbf{x}_i \oplus \widehat{\mathbf{x}}_i$. Then the original text matrix $\mathbf{A}$ is transformed as $\widehat{\mathbf{A}} = \left[\mathbf{x}'_1, \mathbf{x}'_2, \ldots, \mathbf{x}'_i, \ldots, \mathbf{x}'_l\right]^{\mathbf{T}}$, where $\mathbf{x}'_i \in \mathbb{R}^{d+d_s}$ and $\widehat{\mathbf{A}} \in \mathbb{R}^{l\times(d+d_s)}$.

4.2. Multi-view representation

Prior studies primarily focus on the word-level features and do not preserve the association among semantic groups. We introduce the multi-view representation of text to learn both word-level and high-level semantic features including phrase-level and n-gram features. To learn the multi-view representation of textual data, we use the filter in the convolutional layer of CNN to control to what extent the input text is abstracted. Given a filter with filter size k, the convolution operation will generate one feature map based on subsequent k words (referred as k-gram). By changing the filter size k, the raw text sequence can be expressed in different feature maps. We term the feature maps generated by changing filter sizes as the multi-view representation of text.

Given a set of filters $\Theta = \{g_1(\cdot), g_2(\cdot), \cdots, g_k(\cdot)\}$ and the text matrix $\widehat{\mathbf{A}}$, the convolutional operations build the multi-view representation of k-grams by the dot product between a filter $\mathbf{W}_{g_i}$ and the vectors of consecutive words

Table 1
Definition of notations used in this paper.

Notation	Definition
$\mathbf{x}_i$	the word vector of i-th word in the text sequence
l	length of text sequence
d	the dimension of word vector
$\mathbf{A}$	text matrix consisted of word vectors
$\alpha_{i,j}$	attention score between word pair $\mathbf{x}_i$ and $\mathbf{x}_j$
$\boldsymbol{\alpha}$	context attention
$\widehat{\mathbf{x}}_i$	modulated word vector based on attention score α_{ij}
$\mathbf{x}'_i$	concatenation of $\mathbf{x}_i$ and $\widehat{\mathbf{x}}_i$
$\widehat{\mathbf{A}}$	modulated text matrix based on context attention $\boldsymbol{\alpha}$
d_s	dimension of modulated word vector $\widehat{\mathbf{x}}_i$. $d_s = 0$ means context attention is not introduced;
$\mathbf{W}^{\alpha}$	parameter matrix for attention $\boldsymbol{\alpha}$
$g_i(\cdot)$	the convolutional filter with filter size i
θ	set of convolutional filters
k	the number of convolutional filters in $\boldsymbol{\theta}$; it is identical to the number of views;
$\mathbf{v}_i$	features extracted by filter $g_i(\cdot)$, where $1 \leqslant i \leqslant k$
$\mathbf{V}$	feature map matrix generated by filters in θ
$\mathbf{V}^{(i)}$	feature map matrix on the i-th layer
$\mathbf{u}_i$	padded features extracted by $\mathbf{v}_i$
$\mathbf{U}$	padded feature map of $\mathbf{V}$
$\boldsymbol{\beta}$	view attention in parallel-connected SVA
$\boldsymbol{\gamma}$	spatial attention in parallel-connected SVA
$\mathbf{W}^{\beta}$	parameter matrix for view attention $\boldsymbol{\beta}$
$\mathbf{W}^{\gamma}$	parameter matrix for spatial attention $\boldsymbol{\gamma}$
β_j	view attention score in $\boldsymbol{\beta}$
γ_i	spatial attention score in $\boldsymbol{\gamma}$
v_i	mean pooling of the i-th view
$\widehat{\mathbf{v}}$	compressed representation of k views
$\mathbf{W}_1^{\psi}, \mathbf{W}_2^{\psi}$	parameter matrices for series-connected view attention $\boldsymbol{\psi}$
$\mathbf{W}_1^{\chi}, \mathbf{W}_2^{\chi}$	parameter matrices for series-connected view attention $\boldsymbol{\chi}$
$\widetilde{\mathbf{U}}$	modulated feature map based on heterogeneous attention mechanisms
$f(\cdot)$	a set of attention mechanisms. The subscript of $f(\cdot)$ could be s (spatial attention) or v (view attention); The superscript of $f(\cdot)$ could be se (series connection) or pa (parallel connection);
$\boldsymbol{\lambda}$	a vector of Bernoulli random variables

$\mathbf{x}'_{i:k} = \mathbf{x}'_i \oplus \mathbf{x}'_{i+1} \oplus \cdots \oplus \mathbf{x}'_{i+k-1}$. Thus, the feature map based on filter $g_i(\cdot)$ can be written as $\mathbf{v}_i = \mathbf{W}_{g_i} \cdot \left[\mathbf{x}'_{1:k} \oplus \mathbf{x}'_{i:k} \oplus \cdots \oplus \mathbf{x}'_{l:k}\right]$. Finally, we obtain the multi-view representation of text based on Θ, and the set of feature maps can be denoted as $\mathbf{V} = \{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_i, \ldots, \mathbf{v}_k\}$, where $\mathbf{v}_i$ refers to the feature map generated based on filter $g_i(\cdot) \in \Theta$. As the sizes of $\mathbf{v}_i, 1 \leqslant i \leqslant k$ are different, we need to reshape $\mathbf{V}$ to $\mathbf{U}$, and $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_k]$, where $\mathbf{u}_i \in \mathbb{R}^{(l-k+1)\times(d+d_s)}$ represents the i-th view of the feature map $\mathbf{V}$.

4.3. Parallel-connected spatial and view attention (SVA)

As shown in Fig. 2, the parallel-connected spatial and view attention mechanism aims to learn two vectors of weights: the view attention $\boldsymbol{\beta} \in \mathbb{R}^{1\times k}$ to characterize the importance of feature maps generated by filters, and the spatial attention $\boldsymbol{\gamma} \in \mathbb{R}^{(l-k+1)\times k}$ to quantify the weights of semantic groups in each view.

To learn view attention $\boldsymbol{\beta}$ in parallel-connected SVA, we split the multi-view representation of textual into $\mathbf{u}_1$ and $\mathbf{u}_{2:k} = [\mathbf{u}_2, \ldots, \mathbf{u}_k]$. Here $\mathbf{u}_1$ refers to the feature map generated based on unigram features of tokens by the filter $g_1(\cdot)$; $\mathbf{u}_{2:k}$ denotes the phrase-level feature maps generated by filters $g_i(\cdot), 2 \leqslant i \leqslant k$ respectively. As $\mathbf{u}_1$ extracts the word-level features and $\mathbf{u}_i, 2 \leqslant i \leqslant k$ represent the high-level semantic feature groups, we introduce parallel spatial and view attention to learn the correlation between word-level features and high-level semantic groups.

We apply a multilayer perception (MLP) to learn the mapping between $\mathbf{u}_1$ and $\mathbf{u}_i, 2 \leqslant i \leqslant k$. The generated weight matrix vector $\boldsymbol{\beta} \in \mathbb{R}^{1\times k}$ is formulated by Eq. (3), where $\mathbf{W}^{\beta}$ is the weighted matrix; and $b^{\beta} \in \mathbb{R}^{(d+d_s)\times k}$ is the bias vector. The softmax function is utilized to normalize the distribution of attention weights.

$$\boldsymbol{\beta} = softmax\left(tanh\left(\mathbf{W}^{\beta} \cdot [\mathbf{u}_1 \oplus \mathbf{u}_{2:k}] \cdot \mathbf{W}^{\mathbf{T}_{\beta}} + b^{\beta}\right)\right) \tag{3}$$

The spatial attention aims to learn the weight matrix $\boldsymbol{\gamma} = [\gamma_1, \ldots, \gamma_k]$, where k is the number of convolutional filters. $\gamma_i \in \mathbb{R}^{(l-k+1)\times 1}, 1 \leqslant i \leqslant k$ indicates the weight vector of the i-th view. The spatial attention weight vector γ_i is measured by Eq. (4).

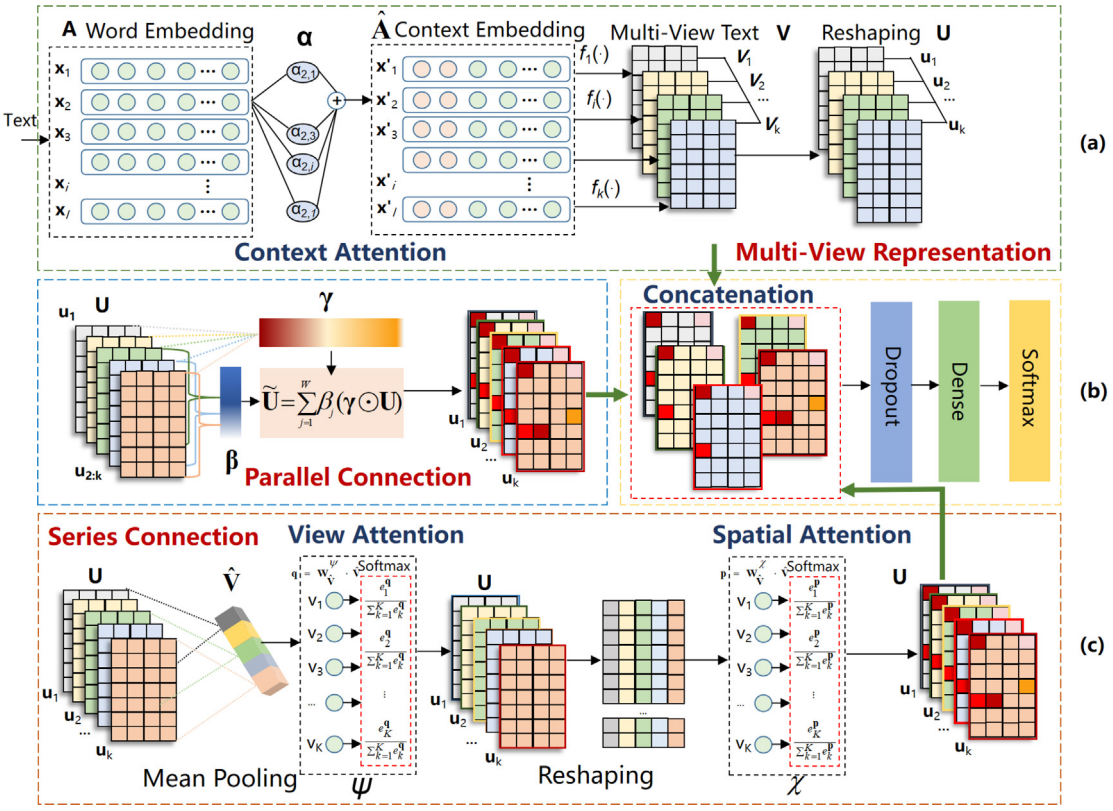

Fig. 2. Proposed SVA-CNN architecture. (a) the multi-view representation extracts the *n*-gram features; (b) parallel attention aims to organize the view and spatial attention mechanisms in the paralleled way; (c) series attention combines the view and spatial attention mechanisms in one series-connected way.

$$\gamma_i = softmax(tanh(\mathbf{W}^{\gamma} \cdot \mathbf{u_i} + b^{\gamma})) \tag{4}$$

Based on the view attention and the spatial attention, the parallel connection of series and parallel attention mechanisms can be formulated as follows, where $\tilde{\mathbf{U}}$ denotes the transformed feature map after parallel SVA.

$$\tilde{\mathbf{U}} = \sum_{j=1}^{W} \beta_j (\boldsymbol{\gamma} \odot \mathbf{U}) \tag{5}$$

4.4. Series-connected spatial and view attention

4.4.1. View attention

After the CNN filter layers, a set of *n*-gram features are generated. However, *n*-grams such as unigram, bi-gram and tri-gram play different roles in the classification task. Therefore, view attention is introduced to learn the more discriminating grams from different views. Specifically, we obtain the structural features $\mathbf{v}_i$ on each view c_i. As the sizes of $\mathbf{v}_i, 1 \leqslant i \leqslant k$ are different, we need to reshape $\mathbf{V}$ to $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_k]$, where $\mathbf{u}_i \in \mathbb{R}^{(l-k+1)\times(d+d_s)}$ represents the i-th view of the feature map $\mathbf{V}$, and k is the number of filters. Then, meanpooling is applied on each view to obtain the feature vector $\hat{\mathbf{v}}$:

$$\hat{\mathbf{v}} = [v_1, v_2, \ldots, v_k]^{\mathbf{T}}, \ \hat{\mathbf{v}} \in \mathbb{R}^{k\times 1} \tag{6}$$

where scalar v_i is the mean of vector $\mathbf{u}_i$, which represents the i-th view features. The view attention mechanism can be formulated as below.

$$\boldsymbol{\psi} = softmax\left(\mathbf{W}_2^{\psi} \cdot tanh\left(\mathbf{W}_1^{\psi} \otimes \hat{\mathbf{v}} + b_1^{\psi}\right) + b_2^{\psi}\right) \tag{7}$$

where $\mathbf{W}_1^{\psi} \in \mathbb{R}^{k\times 1}, \mathbf{W}_2^{\psi} \in \mathbb{R}^{k\times k}$ are parameter matrices, $\otimes$ represents the outer product of vectors. $b_1^{\psi} \in \mathbb{R}^k, b_2^{\psi} \in \mathbb{R}^k$ are the bias terms. The view attention aims to choose the combination of most discriminative *n*-grams for the classification task. Unlike the concatenation of *n*-gram features introduced in [14], the view attention adjusts the weights of *n*-grams according to their contribution to the classifier.

4.4.2. Spatial attention

In general, one word is only highly associated with partial semantic tokens of one sentence. For example, in the theme text classification task, only the semantic phrases associated with the given topic are useful. Therefore, applying a global feature vector to label the whole text may lead to sub-optimal results due to the irrelevant regions. Instead of considering each semantic feature equally, the spatial attention mechanism aims to pay more attention to the semantic-related regions. For the multi-view representation of text $\mathbf{U}$, we reshape $\mathbf{U}$ to $\widehat{\mathbf{U}} = [\widehat{\mathbf{u}}_1, \widehat{\mathbf{u}}_2, \ldots, \widehat{\mathbf{u}}_m]$ by flattening the length and width of $\mathbf{U}$, where $\widehat{\mathbf{u}}_i \in \mathbb{R}^k$, and $m = (l-k+1) \times (d+d_s)$. Thus, the spatial attention $\boldsymbol{\chi}$ can be measured by a single-layer neural network followed by a softmax function.

$$\boldsymbol{\chi} = softmax\Big(\mathbf{W}_2^{\chi} \cdot tanh\Big(\Big(\mathbf{W}_1^{\chi} \cdot \widehat{\mathbf{U}} + b_1^{\chi}\Big)\Big) + b_2^{\chi}\Big) \tag{8}$$

where $\mathbf{W}_1^{\chi} \in \mathbb{R}^{m\times 1}, \mathbf{W}_2^{\chi} \in \mathbb{R}^{m\times m}$ are parameter matrices. $b_1^{\chi} \in \mathbb{R}^m, b_2^{\chi} \in \mathbb{R}^m$ are the bias terms. The spatial attention mechanism aims to learn the phrase-level features associated with the given tokens.

4.4.3. Series connection

With the series collection, we use $\rightarrow$ to indicate the sequential orders of different attention mechanisms. $Ve \rightarrow Sp$ denotes the view attention followed by spatial attention. For the given multi-view inputs of text $\mathbf{U}$, the view attention function F_v aims to obtain the view attention weights $\boldsymbol{\psi}$. After the modulation of $\mathbf{U}$ by $\boldsymbol{\psi}$, we obtain the view guided feature map. Then the modulated feature map is fed to the spatial attention function F_s to obtain the spatial features $\boldsymbol{\chi}$. Finally, we modulate the multi-view representation of text $\mathbf{U}$ by both $\boldsymbol{\psi}$ and $\boldsymbol{\chi}$. The operations of $Ve \rightarrow Sp$ can be formulated as below:

$$\begin{aligned} \boldsymbol{\psi} &= \mathsf{F}_v(\mathbf{U}) \\ \boldsymbol{\chi} &= \mathsf{F}_s\big(f_v^{se}(\mathbf{U}, \boldsymbol{\psi})\big) \\ \widetilde{\mathbf{U}} &= f(\mathbf{U}, \boldsymbol{\psi}, \boldsymbol{\chi}) \end{aligned} \tag{9}$$

Similarly, we define $Sp \rightarrow Ve$ by changing the orders of operations. For $Sp \rightarrow Ve$, given the multi-view representation $\mathbf{U}$, the flatten operation is conducted for $\widehat{\mathbf{U}}$, and the spatial attention function F_s explores the association between tokens. According to $\boldsymbol{\chi}$, view attention function F_v, the operations of $Sp \rightarrow Ve$ can be formulated as below.

$$\begin{aligned} \boldsymbol{\chi} &= \mathsf{F}_s\Big(\widehat{\mathbf{U}}\Big) \\ \boldsymbol{\psi} &= \mathsf{F}_v\big(f_s^{se}(\mathbf{U}, \boldsymbol{\chi})\big) \\ \widetilde{\mathbf{U}} &= f(\mathbf{U}, \boldsymbol{\psi}, \boldsymbol{\chi}) \end{aligned} \tag{10}$$

After the modulation of multi-view textual vector based on spatial and view attention, we obtain one modulated feature map of the initial text representation, denoted as $\widetilde{\mathbf{U}}$.

4.5. Regularization

To overcome the overfitting problem, the dropout layer is employed on the feature vector $\widetilde{\mathbf{U}}$ for regularization. The dropout method randomly skips a proportion of hidden connections among neuron units during the forward training, and ensures the trained model are not tightly dependent on certain set of neuron units. The dropout regularization is shown in Eq. (11), where $\widetilde{\mathbf{U}}$ is the modulated feature vector based on attention mechanisms; $\boldsymbol{\lambda}$ is a vector of Bernoulli random variables with probability p of being 1; $b^{(i)}$ refers the biases of the layer i, and $\widetilde{\mathbb{V}}^{(i)}$ is the output after the dropout operation on the layer i. In addition, the fully connected layer is applied before the softmax classifier. For the FC layer, the activation function is assigned as ReLU.

$$\widetilde{\mathbb{V}}^{(i)} = \mathbf{W}^{(i)}\Big(\widetilde{\mathbf{U}} \cdot \boldsymbol{\lambda}\Big) + b^{(i)} \tag{11}$$

4.6. Loss function

For the data training, we use cross-entropy loss (as shown in Eq. (12)) for the binary classification tasks, where $y_i \in \{0, 1\}$ is the binary indicator, which is 1 for positive samples and 0 for negative samples; $\hat{p}_i$ is the predicted probability of the sample $\mathbf{x}_i$ being positive under the given parameters $\boldsymbol{\theta}$. Here we use sigmod function to build the mapping between sample $\mathbf{x}_i$ and $\hat{p}_i$. While for the multi-class classification problem, we use softmax loss as the loss function. Softmax loss measures the separate cross-entropy loss for each class and minimizes the average loss for all classes.

$$\begin{aligned} loss(\mathbf{x}_i) &= -\frac{1}{n}\sum_{i=1}^{n} y_i \times log(\hat{p}_i) + (1-y_i) \times log(1-\hat{p}_i) \\ &where\ \hat{p}_i = \frac{1}{1+e^{-\theta \mathbf{x}_i}} \end{aligned} \tag{12}$$

To minimize the loss, we optimize this problem by the popular optimizer Adaptive Moment Estimation (Adam). The reasons to choose Adam are threefold. First, it tunes the learning rate during the training automatically. Second, it provides faster convergence compared with stochastic gradient descent algorithm. Finally, it is computationally efficient with less memory requirements.

5. Experiments

In this section, we conduct extensive experiments to answer the following questions:

- **Q1:** Whether the introduction of heterogeneous attention mechanisms is constructive for text classification tasks?
- **Q2:** Which way is efficient to organize the heterogeneous attention mechanisms?
- **Q3:** What parameters are important to tune the model?
- **Q4:** How does the proposed attention series-parallel attention mechanism perform compared with the state-of-the-art models?

5.1. Benchmark datasets

To answer the questions above, we conduct extensive experiment on five public datasets with three classification tasks including document classification, sentiment classification and thematic classification.

- **AG News Corpus (NEWS-4)**[1] [41] consists of news collected from up to 200 sources. All the news are divided into 4 categories based on topics. Each class contains 30,000 training samples and 1,900 testing samples. The total number of training samples is 120,000 and testing 7,600.
- **Hate Speech (HATE-3)**[2] contains 24,784 tweets manually labeled by CrowdFlower users as *hate_speech*, *offensive_language*, or *neither* [42]. As the lack of official split, we randomly choose 21,000 tweets as training samples, and use the rest for testing.
- **Amazon Review (AMZ-5)**[3] consists of reviews of fine foods from Amazon. The dataset includes 500,000 reviews up to October 2012. Reviews include product and user information, ratings, and a plain text review. We use the ratings as the categorial labels, and conduct a 5-class text classification.
- **Movie Review (IMDB-2)**[4] [43] contains up 50,000 movie reviews collected from IMDB.com. It provides a set of 25,000 highly polarized movie reviews for training, and 25,000 for testing. We carry out the binary sentiment classification task to detect the polarized emotion.
- **Emotion Text (TWT-13)**[5] contains labels for 40,000 emotional tweets collected from Twitter. Up to 13 categorial labels are annotated including happiness, sadness, and anger. As the lack of official split, we randomly choose 37,000 tweets as training samples, and use the rest for testing.

5.2. Training setup

The experiments are conducted on a server running RedHat 6.5 operating system with 16-core Intel Xeon E5-2620 CPU @ 2.10 GHz processor, NVIDIA 1080 Ti GPU, and 96 GB RAM. The proposed algorithms in this paper are implemented in Python 3.6 and TensorFlow-GPU 1.9.0. The baselines shown in Section 5.5 are implemented based on the open source package downloaded from Github.[6]

For the CNN model, the dimension of word embedding d is set to 200. For the context attention, the dimension of contextual information d_s is set to 100. In the convolutional layer, we use 5 filters $\Theta = \{g_1(\cdot), g_2(\cdot), g_3(\cdot), g_4(\cdot), g_5(\cdot)\}$ to generate 5 views of text by changing the filter size respectively. Instead of concatenating different views, we apply parallel and series attention mechanisms on multi-view representation to extract the multi-view and multi-granularity attributes for enhanced performance. The proposed solution is publicly available on GitHub.[7]

[1] https://github.com/mhjabreel/CharCNN.
[2] https://github.com/tpawelski/hate-speech-detection.
[3] https://www.kaggle.com/snap/amazon-fine-food-reviews.
[4] http://ai.stanford.edu/amaas/data/sentiment/.
[5] https://www.figure-eight.com/data-for-everyone/.
[6] https://github.com/brightmart/text_classification.
[7] https://github.com/yunjinwpu/SVA-CNN.

5.3. SVA-CNN evaluations

To examine whether the introduction of different attention mechanisms including context attention, spatial attention and view attention is beneficial for the text classification task (**Q1**) and to *identify* the efficient way to organize attention mechanisms (**Q2**), we evaluate the performance of heterogeneous attention mechanisms on five public datasets. For simplicity, Co, Sp, and Ve denote the CNN model with context attention, spatial attention and view attention, respectively. To examine the performance of the combination of three basic attention mechanisms, we introduce two operators: $\rightarrow$ for series connection and // for parallel connection. Based on these two operators, we construct the following combinations: $Co \rightarrow Sp$ for the combination of context attention and spatial attention; $Co \rightarrow Ve$ for the integration of context and view attention; $Sp//Ve$ for paralleled SVA as defined in Eq. (5); $Sp \rightarrow Ve$ and $Ve \rightarrow Sp$ are short for series connection of spatial and view attention mechanisms in different sequential orders. $Co \rightarrow (Sp//Ve)$ denotes for the paralleled view and spatial attention with the contextual embedding; $Co \rightarrow (Sp \rightarrow Ve)$ and $Co \rightarrow (Ve \rightarrow Sp)$ represent the series connection of three basic attention mechanisms in different orders. We evaluate the performance of heterogeneous attention mechanisms on *five* public datasets respectively. Based on the results presented in Table 2, we make the following key observations:

1) Compared with the contextual attention Co, introducing the spatial attention Sp and view attention Ve improves performance on all datasets, and the spatial attention Sp performs best among the three basic attention mechanisms. The spatial attention Sp outperforms context attention with over 7% relative gain on all datasets. The context attention primarily uses word level features, while spatial attention and view attention mechanisms capture high-level features such as phrases and semantics. The powerful capability of representation learning for complex features enables both spatial and view attention mechanism to gain significant improvements for text classification.
2) Compared with single attention mechanism, heterogeneous attention mechanisms enhance performance. For instance, the series connection of context attention Co and spatial attention Sp shows significant performance gain in terms of accuracy. This holds true no matter how the heterogeneous attention mechanisms are connected. This finding indicates that the mixture of attention mechanisms learns supplementary features for text classification. Furthermore, the connection of heterogeneous attention mechanisms matters with regard to the overall performance. In our experiments, we conduct a comparative study to show differences between series connection and parallel connection. According to Table 2, we observe that series-connected spatial and view attention mechanisms $Sp \rightarrow Ve$ and $Ve \rightarrow Sp$ outperform other combinations. In addition, there are subtle differences between $Sp \rightarrow Ve$ and $Ve \rightarrow Sp$. In general, the series connection of $Ve \rightarrow Sp$ performs better compared with $Sp \rightarrow Ve$.
3) We investigate the effects of context attention on the combinations of spatial attention and view attention mechanisms. As shown in Table 2, $Ve \rightarrow Sp$ performs well with the accuracy of 97.46%; while $Co \rightarrow (Ve \rightarrow Sp)$ has 97.25%; $Ve//Sp$ outperforms $Co \rightarrow (Ve//Sp)$ with 0.82% gain in terms of accuracy on NEWS-4. This indicates that context attention Co is not essential and can be replaced with the combination of spatial attention Sp and view attention Ve. This indicates that $Ve \rightarrow Sp$ and $Ve//Sp$ can learn the long-term dependencies.

5.4. Impacts of multi-view representation

To answer **Q3:** *what parameters are important to tune the model?*, we evaluate the impacts of multiple views and padding length, respectively. With regard to the multi-view representation, we study the effects of multiple views on classification performance by changing the number of k, which refers to the number of views and is determined by the number of filters defined in Θ (See Section 4.2). Here, we set $k = 3$, 6, and 9 respectively, and examine the corresponding performance of $Ve \rightarrow Sp, Sp//Ve$ and $Sp \rightarrow Ve$.

According to Table 3, we observe that the performance of series/ parallel-connected SVA is enhanced with the increase of k. For example, when $k = 3$, the accuracy of $Sp//Ve$ is 95.30%; while it is 96.98% when $k = 9$. The increase of view number k indicates that multi-granularity semantic groups are extracted by leveraging multi-view representation learning of textual data, which can preserve fine-grained semantic features. It also demonstrates that introducing additional views increases overall performance.

Since how many views should be introduced is still an open question, we track the performance of $Ve \rightarrow Sp$, and $Sp \rightarrow Ve$ on NEWS-4 by changing the number of views k. As shown in Fig. 3, generally the performance of SVA is enhanced with the increase of views when $k \leqslant 6$. While, the training time consumption increases when including additional views. For example, when $k \leqslant 5$, the performance improvement is significant with lower training time consumption. While when $k \geqslant 6$, the accuracy gain is subtle with the increase of views. In contrast, the training time grows rapidly. Thus, the number of views should be limited due to the trade-off between performance and training time consumption. To balance the time consumption and the accuracy, we set the number of view $k = 5$.

In addition to examining the sensitivity of these parameters on the overall performance, we study the impact of padding length l on performance. The padding length is widely applied to format the raw data into a fixed-length vector. In the comparative experiment, we set the padding length $l \in \{100, 200, 300, 400, 500\}$ respectively. As shown in Fig. 4, although the combination of context, view and spatial attention mechanisms $Co \rightarrow (Ve \rightarrow Sp)$ performs best in terms of accuracy, $Ve \rightarrow Sp$ and $Sp \rightarrow Ve$ show competitive performance in terms of accuracy against $Co \rightarrow (Ve \rightarrow Sp)$ with the increase of pad-

Table 2
Performance of different connections of contextual, spatial, and view attention mechanisms.

Dataset	Model	Metrics (%)			
		Acc	*F1*	*P*	*R*
NEWS-4	*Co*	84.38	80.56	90.63	85.30
	Sp	92.23	92.57	92.42	92.36
	Ve	90.63	89.24	90.43	90.38
	$Co \rightarrow Sp$	95.45	95.27	95.24	95.50
	$Co \rightarrow Ch$	93.76	93.90	93.32	93.45
	$Sp//Ch$	96.12	96.47	96.82	96.19
	$Sp \rightarrow Ve$	**97.44**	**97.69**	**97.81**	**97.28**
	$Ve \rightarrow Sp$	**97.46**	**97.65**	**97.78**	**97.41**
	$Co \rightarrow (Sp//Ve)$	95.30	93.70	92.52	96.90
	$Co \rightarrow (Sp \rightarrow Ve)$	97.25	93.41	93.15	97.09
	$Co \rightarrow (Ve \rightarrow Sp)$	97.12	93.29	93.73	97.31
HATE-3	*Co*	84.51	80.69	90.76	85.43
	Sp	92.36	92.70	92.55	92.49
	Ve	90.76	89.37	90.56	90.51
	$Co \rightarrow Sp$	95.58	95.40	95.37	95.63
	$Co \rightarrow Ve$	93.89	94.03	93.45	93.58
	$Sp//Ve$	96.25	96.60	96.95	96.32
	$Sp \rightarrow Ve$	**97.57**	**97.82**	**97.94**	**97.41**
	$Ve \rightarrow Sp$	**97.59**	**97.78**	**97.91**	**97.54**
	$Co \rightarrow (Sp//Ve)$	94.30	92.70	91.52	95.90
	$Co \rightarrow (Sp \rightarrow Ve)$	96.25	92.41	92.15	96.09
	$Co \rightarrow (Ve \rightarrow Sp)$	96.12	92.29	92.73	96.31
AMZ-5	*Co*	84.27	80.45	90.52	85.19
	Sp	92.12	92.46	92.31	92.25
	Ve	90.52	89.13	90.32	90.27
	$Co \rightarrow Sp$	95.34	95.16	95.13	95.39
	$Co \rightarrow Ve$	93.65	93.79	93.21	93.34
	$Sp//Ve$	96.01	96.36	96.71	96.08
	$Sp \rightarrow Ve$	**97.33**	**97.58**	**97.70**	**97.17**
	$Ve \rightarrow Sp$	**97.35**	**97.54**	**97.67**	**97.30**
	$Co \rightarrow (Sp//Ve)$	96.60	94.80	93.92	98.10
	$Co \rightarrow (Sp \rightarrow Ve)$	98.55	94.51	94.55	98.29
	$Co \rightarrow (Ve \rightarrow Sp)$	98.42	94.39	95.13	98.51
IMDB-2	*Co*	85.08	81.26	91.33	86.00
	Sp	92.93	93.27	93.12	93.06
	Ve	91.33	89.94	91.13	91.08
	$Co \rightarrow Sp$	96.15	95.97	95.94	96.20
	$Co \rightarrow Ve$	94.46	94.60	94.02	94.15
	$Sp//Ve$	96.82	97.17	97.52	96.89
	$Sp \rightarrow Ve$	**98.14**	**98.39**	**98.51**	**97.98**
	$Ve \rightarrow Sp$	**98.16**	**98.35**	**98.48**	**98.11**
	$Co \rightarrow (Sp//Ve)$	95.81	94.21	93.03	97.01
	$Co \rightarrow (Sp \rightarrow Ve)$	97.76	93.92	93.66	97.20
	$Co \rightarrow (Ve \rightarrow Sp)$	97.63	93.80	94.24	97.42
TWT-13	*Co*	86.18	82.36	92.43	87.10
	Sp	94.03	94.37	94.22	94.16
	Ve	92.43	91.04	92.23	92.18
	$Co \rightarrow Sp$	97.25	97.07	97.04	97.30
	$Co \rightarrow Ve$	95.56	95.70	95.12	95.25
	$Sp//Ve$	97.92	98.27	98.62	97.99
	$Sp \rightarrow Ve$	**99.24**	**99.49**	**99.61**	**99.08**
	$Ve \rightarrow Sp$	**99.26**	**99.45**	**99.58**	**99.21**
	$Co \rightarrow (Sp//Ve)$	96.19	94.59	93.41	97.39
	$Co \rightarrow (Sp \rightarrow Ve)$	98.14	94.30	94.04	97.58
	$Co \rightarrow (Ve \rightarrow Sp)$	98.01	94.18	94.62	97.80

ding length l. This indicates that increasing the padding length can help improve accuracy. A closer examination of the results also indicates that the training time consumption of $Co \rightarrow (Ve \rightarrow Sp)$ shows exponential growth with the increase of padding length l. Compared with the high training time of $Co \rightarrow (Ve \rightarrow Sp)$, $Ve \rightarrow Sp$ and $Sp \rightarrow Ve$ have lower training time consumption. When examining the trade off between the performance and training time consumption, $Ve \rightarrow Sp$ and $Sp \rightarrow Ve$ show competitive performance and lower training consumption.

Table 3
Effects of multi-view representation on heterogeneous attention mechanisms.

Dataset	*Attention*	*k*	Metrics (%)			
			Acc	*F1*	*P*	*R*
NEWS-4	*Sp//Ve*	3	95.30	93.70	92.52	96.90
		6	96.85	93.16	92.77	96.91
		9	96.98	93.62	92.88	96.97
	Sp → Ve	3	97.25	93.41	93.15	97.09
		6	97.30	93.37	93.33	97.18
		9	97.41	93.33	93.53	97.44
	Ve → Sp	3	97.12	93.29	93.73	97.31
		6	97.32	93.25	93.86	97.12
		9	97.47	93.21	94.13	97.82
HATE-3	*Sp//Ve*	3	94.30	92.70	91.52	95.90
		6	95.85	92.16	91.77	95.86
		9	95.98	92.62	91.88	95.99
	Sp → Ve	3	96.25	92.41	92.15	96.09
		6	96.30	92.37	92.33	96.18
		9	96.41	92.33	92.53	96.44
	Ve → Sp	3	96.12	92.29	92.73	96.31
		6	96.32	92.25	92.86	96.12
		9	96.47	92.21	93.13	96.82
AMZ-5	*Sp//Ve*	3	96.60	94.80	93.92	98.10
		6	98.15	94.26	94.17	98.26
		9	98.28	94.72	94.28	98.19
	Sp → Ve	3	98.55	94.51	94.55	98.29
		6	98.60	94.47	94.73	98.38
		9	98.71	94.43	94.93	98.64
	Ve → Sp	3	98.42	94.39	95.13	98.51
		6	98.62	94.35	95.26	98.32
		9	98.77	94.31	95.53	99.02
IMDB-2	*Sp//Ve*	3	95.81	94.21	93.03	97.01
		6	97.36	93.67	93.28	97.24
		9	97.49	94.13	93.39	97.10
	Sp → Ve	3	97.76	93.92	93.66	97.20
		6	97.81	93.88	93.84	97.29
		9	97.92	93.84	94.04	97.55
	Ve → Sp	3	97.63	93.80	94.24	97.42
		6	97.83	93.76	94.37	97.23
		9	97.98	93.72	94.64	97.93
TWT-13	*Sp//Ve*	3	96.19	94.59	93.41	97.39
		6	97.74	94.05	93.66	97.64
		9	97.87	94.51	93.77	97.48
	Sp → Ve	3	98.14	94.30	94.04	97.58
		6	98.19	94.26	94.22	97.67
		9	98.30	94.22	94.42	97.93
	Ve → Sp	3	98.01	94.18	94.62	97.80
		6	98.21	94.14	94.75	97.61
		9	98.36	94.10	95.02	98.31

5.5. Summary of benchmark experiments

To answer **Q4**, we compare series-connected spatial and view attention mechanisms *Ve → Sp* with state-of-the-art deep learning-based classification algorithms. They are as follows:

- **TextCNN** was first proposed for text classification in [14], where the weights of words are assigned according to the pre-trained vectors, and can be dynamically tuned during training.
- **TextRNN**: LSTM and GRU are widely applied for sequence learning task.In this paper, we choose the bidirectional LSTM (Bi-LSTM) as the baseline.
- **CLSTM** [44] utilizes CNN to extract the higher-level phrase representations, and employs LSTM to obtain the sentence representation.CLSTM is able to capture both local features of phrases as well as global and temporal sentence semantics.
- **RCNN** [45] applies a bi-directional recurrent structure to capture contextual information as far as possible when learning word representations, and utilizes a max-pooling layer to capture the key clues in text.

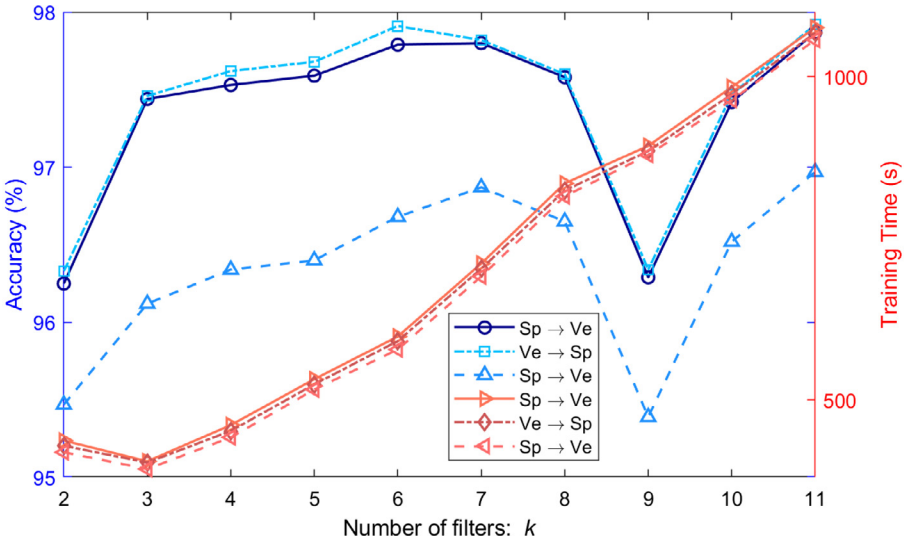

Fig. 3. Changes of performance with the increase of views k. Blue lines show the performance in accuracy; red lines summarize training time.

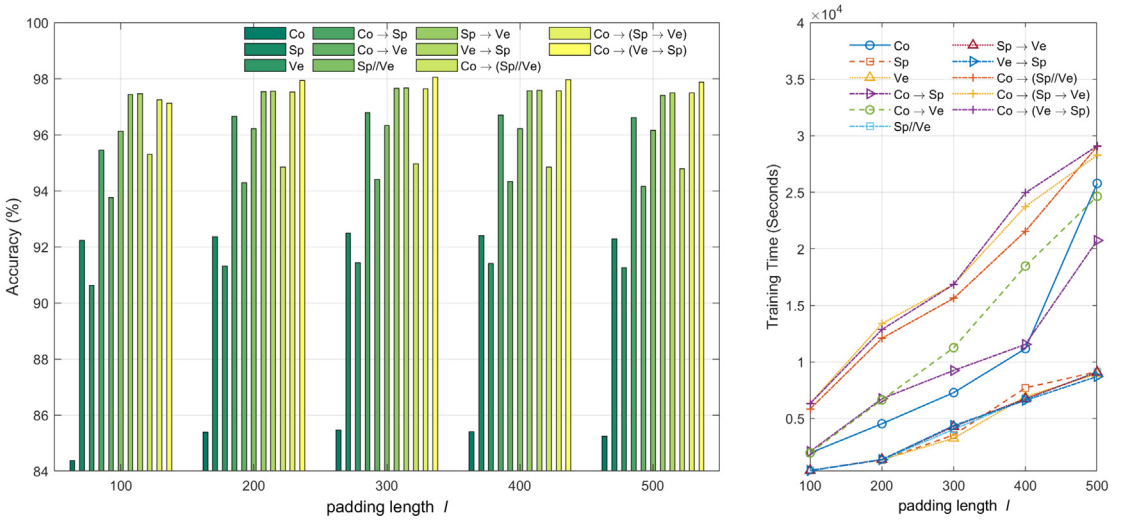

Fig. 4. Performance changes and training time consumption with the increase of padding length l among the combinations of heterogeneous attention mechanisms.

- **Seq2seq** [17] provides one unified end-to-end framework for sequence learning tasks.It relies on a multilayered LSTM to map the input sequence to a vector of a fixed dimensionality, and then another LSTM model to decode the target sequence.
- **Hierarchical Attention Networks** (HAN) explores the hierarchical structure of documents and provides two attention mechanisms to learn the word-level and sentence-level representations of documents [38].
- **FastText** [46] utilizes the bag-of-words of n-gram features instead of word vectors to capture partial information about the local word order.Then the averaged text representation is fed to a hierarchical softmax to speed up the training process.
- **Self-Attention** [47] aims to compute the weight for each vector by comparing itself with all vectors.With the self-attentive mechanism, the network learns the latent interaction among tokens and adjust the weights of tokens correspondingly.
- **DPCNN** [48] integrates the region embedding and shortcut connection in low-complexity word-level CNN for text classification to efficiently represent long-range associations in text.

- **Transformer** [40] utilizes self-attention of words, positional embedding and multi-head attention to long-term dependency. Transformer is independent from RNN and CNN models, and is highly parallelism with low computational complexity.

As shown in Table 4, we find that CNN-based models show competitive or even better results than that of RNN-based models. This demonstrates that CNN can be used to sequence learning tasks classification tasks as well, which is consistent with the conclusions in prior studies [14]. In addition, the combination of RNN and CNN such as CLSTM and RCNN does not significantly improve the overall performance.

Compared with models without attention mechanisms (e.g., TextCNN, RCNN, CLSTM, RCNN, DPCNN), the models with attention mechanisms including HAN, self-attention, and transformer outperform the models without attention mechanisms on all five datasets. This indicates that attention mechanisms can significant improve the classification performance by modulating the weights of features.

We also conduct statistical significance tests according to the results shown in Table 4 to quantify the differences among different models. Specifically, we compare the metrics of baselines with that of $Ve \rightarrow Sp$ respectively. Apart from the results on dataset *IMDB-2*, significant differences are observed with the value of $p < 0.05$. For the dataset *IMDB-2*, significant differences exist among $Ve \rightarrow Sp$ and other baselines except transformer. These observations indicate $Ve \rightarrow Sp$ yields a significant performance gain compared to other classification methods. For example, the averaged accuracy of $Ve \rightarrow Sp$ on all five public datasets is over 96%. The strong performance of $Ve \rightarrow Sp$ indicates that the proposed connection of spatial and view attention for multi-view textual data in CNN effectively learns the latent dependency and long-term dependency for text classification tasks. In contrast to the spatial attention in HAN, self-attention, and transformer, $Ve \rightarrow Sp$ preserves the long-term word-level dependency, and the high-level semantic features.

Since the CNN relies on padding formulations, we also conduct additional evaluations to evaluate the performance of baselines and $Ve \rightarrow Sp$ with the increase of padding length ranging l ranging from 100 to 500. Conducting such a sensitivity analysis is critical to identifying optimal and peak performances of CNN-based models operating in text contexts. For simplicity and space considerations, we choose NEWS-4 as the training dataset. As shown in Fig. 5, $Ve \rightarrow Sp$ outperforms the other baselines. In terms of training time consumption, the time costs for training on NEWS-4 grow steadily for baselines with the increase of padding length. Among the baselines, Transformer shows the highest time consumption. We compare our proposed solution with transformer in terms of efficiency based on the training time. We observe that when $l \in \{100, 200\}, Ve \rightarrow Sp$ outperforms all the baselines with highest accuracy and least training time consumption. Although the training time consumption grows rapidly with the increase of padding length, $Ve \rightarrow Sp$ shows competitive performance than that of transformer with similar time consumption and a significant grain margin in terms of accuracy.

6. Discussion

Compared with prior works in the field of text classification, our proposed solutions have four key advantages. First, the five integrated attention mechanisms in the novel attention pool are diverse and carefully selected to address common scenarios that occur within text classification. The spatial attention characterizes the long-term dependencies among n-gram features by quantifying the importance of n-gram features for text classification. The view attention mechanism quantifies which channel of the n-gram features are important. The series-parallel connection of heterogeneous attention mechanisms reveals the efficient way for multi-view text classification. Second, the attention mechanisms help to improve the performance of the proposed approaches by carefully considering key characteristics within input text. Third, no manual feature engineering is required at any stage during the process. This is an essential functionality for rapidly evolving domains. Finally, the method outputs interpretable features that are weighted based on importance to the final decision making process. Consequently, it can be used for subsequent downstream machine learning tasks, enhance domain value, and help to refine the model for future applications.

To illustrate the advantages and potential practical utility of SVA-CNN for text classification, we visualize the spatial and view attention on selected messages from NEWS-4. The original text is depicted as below:

A suicide car bomb detonated outside the gates of a Marine base in western Iraq on Saturday killing at least sixteen Iraqi police officers and wounding forty others at a police checkpoint. Afghan police are investigating a suicide grenade attack in the center of Kabul that injured seven people including three international peacekeepers. An American woman has died after being wounded by a suicide bomber in Kabul. The US military said taking the death toll from the attack on a crowded shopping street popular with foreigners to two. Kabul bomb deaths rise to three. Afghan authorities say a US translator has died after a suicide bombing in Kabul bringing the number killed to three. An American woman and an Afghan girl died from wounds suffered in a Taliban suicide attack in a popular Kabul shopping street. US embassy and hospital officials said on Sunday an American woman and a young Afghan girl have died from their wounds after a suicide bomb attack in the Afghan capital of Kabul. A Taliban suicide fighter blew himself up in the attack.

Fig. 6 shows effects of heterogeneous attention mechanisms in SVA-CNN. The horizontal axis are the indexing number of words in the given example, and represent the spatial attention weights on the given view. The color depth expresses the importance degree of one word in the spatial attention mechanism. The more important role the word plays in semantic

Table 4
Comparison of the proposed SVA-CNN against State-of-the-Art Benchmark Algorithms.

Dataset	Model	Metrics (%)			
		Acc	*F*1	*P*	*R*
NEWS-4	TextCNN	83.58	78.66	83.67	74.94
	TextRNN	84.67	83.66	83.68	83.99
	CLSTM	84.59	79.67	84.68	75.95
	RCNN	87.63	83.70	87.71	79.99
	Seq2Seq	86.60	80.69	86.70	76.96
	HAN	91.76	88.68	85.75	93.04
	FastText	90.75	89.69	86.74	92.05
	Self-Attention	89.74	88.69	86.73	91.04
	DPCNN	81.63	80.64	81.65	79.96
	Transformer	92.75	91.75	92.76	92.07
	SVA-CNN	**97.46**	**97.65**	**97.78**	**97.41**
HATE-3	TextCNN	88.72	87.69	86.72	89.03
	TextRNN	89.74	86.64	81.73	91.02
	CLSTM	90.70	88.73	90.74	87.04
	RCNN	89.74	87.68	85.73	91.03
	Seq2Seq	90.70	89.74	91.74	87.05
	HAN	94.77	93.77	94.78	94.09
	FastText	95.76	94.78	95.79	93.10
	Self-Attention	94.75	93.78	95.78	92.09
	DPCNN	90.73	90.73	90.74	90.06
	Transformer	94.79	95.78	95.78	96.11
	SVA-CNN	**97.59**	**97.78**	**97.91**	**97.54**
AMZ-5	TextCNN	87.62	86.78	95.71	79.02
	TextRNN	91.74	91.75	92.75	91.07
	CLSTM	89.69	88.73	90.73	86.04
	RCNN	92.68	90.81	98.76	85.06
	Seq2Seq	85.63	82.68	85.69	79.98
	HAN	92.74	91.75	92.76	91.07
	FastText	91.75	92.74	91.75	92.08
	Self-Attention	92.76	92.74	91.76	93.08
	DPCNN	88.73	84.63	80.72	90.00
	Transformer	94.79	93.75	92.78	96.09
	SVA-CNN	**97.35**	**97.54**	**97.67**	**97.30**
IMDB-2	TextCNN	92.77	93.76	93.76	94.09
	TextRNN	93.78	93.74	91.77	95.09
	CLSTM	91.70	90.76	93.75	87.06
	RCNN	95.80	95.76	93.79	97.11
	Seq2Seq	92.64	87.78	95.76	81.03
	HAN	94.81	93.72	89.78	98.09
	FastText	94.78	93.76	93.78	95.09
	Self-Attention	95.76	95.81	98.79	93.11
	DPCNN	92.77	91.73	90.76	94.07
	Transformer	97.82	97.79	96.81	99.13
	SVA-CNN	**98.16**	**98.35**	**98.48**	**98.11**
TWT-13	TextCNN	90.61	84.75	92.74	78.00
	TextRNN	85.60	81.70	87.69	76.97
	CLSTM	87.65	85.73	90.71	82.01
	RCNN	86.63	85.75	92.70	80.01
	Seq2Seq	85.50	78.77	94.69	66.94
	HAN	86.81	74.42	59.70	97.90
	FastText	85.70	83.67	84.69	86.99
	Self-Attention	89.69	90.79	96.73	86.06
	DPCNN	91.75	90.72	89.75	92.06
	Transformer	93.78	93.76	93.77	95.09
	SVA-CNN	**99.26**	**99.45**	**99.58**	**99.21**

representation, the darker the color is. For the vertical axis, it enumerates the views generated by different convolutional filters, and aims to show the significance of views for the classification task.

As illustrated in Fig. 6, spatial attention distributions of multiple views are significantly different. For example, on word-level features, the words such as *suicide, car, bomb* are assigned with higher weight scores. While the for *n*-gram feature, more high-level semantic groups are taken as the important clues for text classification. For instance, *suicide bomber*, *Iraqi police officers* and *suicide grenade attack* are assigned with higher spatial weights. Meanwhile, according to the view weights, the view attention mechanism can quantify the importance of multiple views for the given task. According to the visualiza-

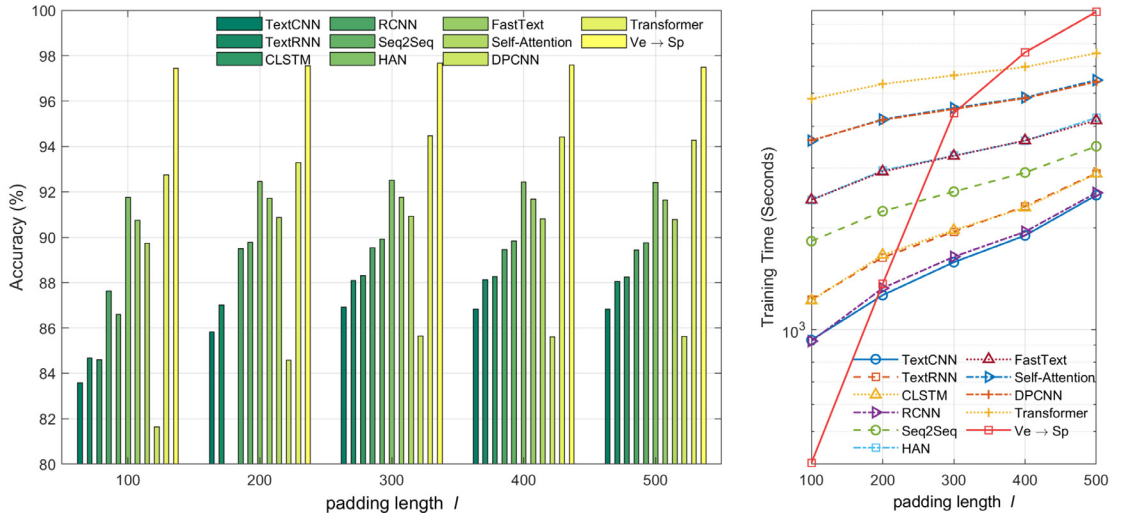

Fig. 5. Performance changes and training time consumption with the increase of padding length l between $Ve \rightarrow Sp$ and baselines.

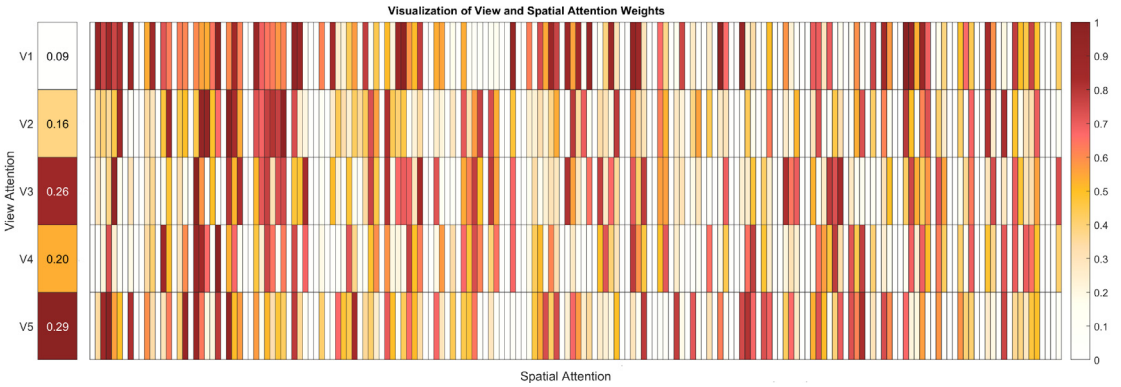

Fig. 6. Visualization of spatial and view attention mechanisms. In this figure, each row indicates the spatial attention weights for one given view; the columns illustrate the weight distribution among views.

tion of spatial and view attention mechanisms, we find that SVA-CNN can provide the multiple granularity feature maps through multi-view representation learning, and tune the weights of multi-view features for enhanced performance.

7. Conclusion

Despite the prevalence of text classification studies, existing solutions are labor-intensive and task-oriented feature engineering of conventional text classification methods are not suitable for the rapidly evolving domains. Moreover, prevailing deep-learning based solutions only represent text as a single view and omit multiple sets of features at varying levels of granularity. To date, encoding different levels of granularity to enrich fine-grained text representation at semantic level is rarely studied. In this paper, we propose a novel Spatial View Attention Convolutional Neural Network (SVA-CNN). SVA-CNN leverages the multi-view representation of sequential text and a heterogeneous combination of context, spatial and view attention mechanisms to automatically extract and weight multiple granularities and fine-grained representations. The proposed SVA-CNN is evaluated on five datasets with document classification, sentiment classification, and thematic classification tasks. The experimental results show that SVA-CNN outperforms baselines by learning the multi-granularity features from multi-view text representation, and the series-connected spatial and view attention mechanism yields a significant performance gain in all tasks with higher accuracy and lower training time.

CRediT authorship contribution statement

Yunji Liang: conceptualization, formal analysis, investigation, methodology, writting - original draft. **Huihui Li:** data curation, software, validation. **Bin Guo:** project administration, writing- review&editing. **Zhiwen Yu:** project administration,

writing- review&editing. **Xiaolong Zheng:** funding acquisition, project administration, writing-review & editing. **Sagar Samtani:** visualization, writing (reviewing & edit). **Daniel Zeng:** supervision, writing (review&editing).

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work is supported by the National Key Research and Development Program of China under Grant No.: 2019YFB2102200, by the ministry of health of China under Grant No.: 2017ZX10303401-002 and 2017YFC1200302, by the natural science foundation of China under Grant No.: 61902320, 71472175, 71602184, 71621002, by national science foundation under Grant No. CNS-1850362 and OAC-1917117, and by the fundamental research funds for the central universities under Grant No.:31020180QD140.

References

[1] M. Dragoni, An evolutionary strategy for concept-based multi-domain sentiment analysis, IEEE Comput. Intell. Mag. 14 (2) (2019) 18–27.
[2] L. Kong, C. Li, J. Ge, F. Zhang, Y. Feng, Z. Li, B. Luo, Leveraging multiple features for document sentiment classification, Inf. Sci. 518 (2020) 39–55.
[3] T.K. Mackey, J. Kalyanam, T. Katsuki, G. Lanckriet, Twitter-based detection of illegal online sale of prescription opioid, Am. J. Public Health 107 (12) (2017) 1910–1915.
[4] T. Mackey, J. Kalyanam, J. Klugman, E. Kuzmenko, R. Gupta, Solution to detect, classify, and report illicit online marketing and sales of controlled substances via twitter: using machine learning and web forensics to combat digital opioid access, J. Med. Internet Res. 20 (4) (2018), e10029.
[5] Y. Liang, X. Zheng, D.D. Zeng, A survey on big data-driven digital phenotyping of mental health, Inf. Fusion 52 (2019) 290–307.
[6] J.C. Eichstaedt, R.J. Smith, R.M. Merchant, L.H. Ungar, P. Crutchley, D. Preoţiucpietro, D.A. Asch, H.A. Schwartz, Facebook language predicts depression in medical records, Proc. Natl. Acad. Sci. U.S.A. 115 (44) (2018) 11203–11208.
[7] J.C.S. Reis, A. Correia, F. Murai, A. Veloso, F. Benevenuto, E. Cambria, Supervised learning for fake news detection, IEEE Intell. Syst. 34 (2) (2019) 76–81.
[8] D. Lazer, M.A. Baum, Y. Benkler, A.J. Berinsky, K.M. Greenhill, F. Menczer, M.J. Metzger, B. Nyhan, G. Pennycook, D. Rothschild, et al, The science of fake news, Science 359 (6380) (2018) 1094–1096.
[9] S. Hassanpour, N. Tomita, T. Delise, B.S. Crosier, L.A. Marsch, Identifying substance use risk based on deep neural networks and instagram social media data, Neuropsychopharmacology 44 (3) (2019) 487–494.
[10] T. Wang, M. Brede, A. Ianni, E. Mentzakis, Detecting and characterizing eating-disorder communities on social media, in: Proceedings of the Tenth ACM International Conference on Web Search and Data Mining, ACM, 2017, pp. 91–100..
[11] S. Sun, C. Luo, J. Chen, A review of natural language processing techniques for opinion mining systems, Inf. Fusion 36 (2017) 10–25.
[12] R. Ren, D.D. Wu, T. Liu, Forecasting stock market movement direction using sentiment analysis and support vector machine, IEEE Syst. J. 13 (1) (2019) 760–770.
[13] R. Alzaidy, C. Caragea, C. L. Giles, Bi-lstm-crf sequence labeling for keyphrase extraction from scholarly documents, in: Proceedings of the World Wide Web Conference (WWW), Association for Computing Machinery, New York, NY, USA, 2019, pp. 2551–2557..
[14] Y. Kim, Convolutional neural networks for sentence classification, in: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics, Doha, Qatar, 2014, pp. 1746–1751..
[15] Q. Cheng, T.M.H. Li, C.-L. Kwok, T. Zhu, P.S.F. Yip, Assessing suicide risk and emotional distress in chinese social media: a text mining and machine learning study, J. Med. Internet Res. 19 (7) (2017), e243.
[16] R. Xia, C. Zong, S. Li, Ensemble of feature sets and classification algorithms for sentiment classification, Inf. Sci. 181 (6) (2011) 1138–1152.
[17] I. Sutskever, O. Vinyals, Q. V. Le, Sequence to sequence learning with neural networks, in: Annual Conference on Neural Information Processing Systems 2014, December 8–13 2014, Montreal, Quebec, Canada, 2014, pp. 3104–3112..
[18] J. Kim, Y. Ko, J. Seo, A bootstrapping approach with crf and deep learning models for improving the biomedical named entity recognition in multi-domains, IEEE Access 7 (2019) 70308–70318.
[19] Y. Zhang, Z. Zhang, D. Miao, J. Wang, Three-way enhanced convolutional neural networks for sentence-level sentiment classification, Inf. Sci. 477 (2019) 55–64.
[20] X. Li, Y. Rao, H. Xie, X. Liu, T.-L. Wong, F.L. Wang, Social emotion classification based on noise-aware training, Data & Knowl. Eng. 123 (2019), 101605.
[21] M. Huang, H. Xie, Y. Rao, J. Feng, F.L. Wang, Sentiment strength detection with a context-dependent lexicon-based convolutional neural network, Inf. Sci. 520 (2020) 389–399.
[22] S. K. Sahu, F. Christopoulou, M. Miwa, S. Ananiadou, Inter-sentence relation extraction with document-level graph convolutional neural network, in: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Florence, Italy, 2019, pp. 4309–4316..
[23] L. Wang, Z. Cao, G. de Melo, Z. Liu, Relation classification via multi-level attention cnns, in: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, Berlin, Germany, 2016, pp. 1298–1307..
[24] E. Strubell, A. Ganesh, A. McCallum, Energy and policy considerations for deep learning in NLP, in: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Florence, Italy, 2019, pp. 3645–3650..
[25] J. Zhao, X. Xie, X. Xu, S. Sun, Multi-view learning overview: recent progress and new challenges, Inf. Fusion 38 (2017) 43–54.
[26] Y. Li, M. Yang, Z. Zhang, A survey of multi-view representation learning, IEEE Trans. Knowl. Data Eng. 31 (10) (2019) 1863–1883.
[27] M. Du, F. Li, G. Zheng, V. Srikumar, Deeplog: anomaly detection and diagnosis from system logs through deep learning, in, in: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, 2017, pp. 1285–1298.
[28] S. Hassanpour, N. Tomita, T. DeLise, B. Crosier, L.A. Marsch, Identifying substance use risk based on deep neural networks and instagram social media data, Neuropsychopharmacology 44 (2019) 487–494.
[29] S. Chancellor, Y. Kalantidis, J.A. Pater, M. De Choudhury, D.A. Shamma, Multimodal classification of moderated online pro-eating disorder content, in: Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems, Denver, Colorado, USA, 2017, pp. 3213–3226..
[30] B. Jiang, Z. Li, H. Chen, A.G. Cohn, Latent topic text representation learning on statistical manifolds, IEEE Trans. Neural Netw. Learn. Syst. 29 (11) (2018) 5643–5654.
[31] L. Niu, W. Li, D. Xu, J. Cai, An exemplar-based multi-view domain generalization framework for visual recognition, IEEE Trans. Neural Netw. Learn. Syst. 29 (2) (2018) 259–272.
[32] T. Zhang, W. Zheng, Z. Cui, Y. Zong, J. Yan, K. Yan, A deep neural network-driven feature learning method for multi-view facial expression recognition, IEEE Trans. Multimedia 18 (12) (2016) 2528–2536.

[33] L. Qiu, Y. Zhang, P. Jin, H. Wang, Multi-view Chinese treebanking, in: 25th International Conference on Computational Linguistics, Dublin City University and Association for Computational Linguistics, Dublin, Ireland, 2014, pp. 257–268..
[34] P.S. Dhillon, D.P. Foster, L.H. Ungar, Multi-view learning of word embeddings via CCA, in, in: Proceedings of the 25th Annual Conference on Neural Information Processing Systems, 2011, pp. 199–207.
[35] K. Cho, B. van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, Y. Bengio, Learning phrase representations using RNN encoder–decoder for statistical machine translation, in: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics, Doha, Qatar, 2014, pp. 1724–1734..
[36] Z. Zhao, Y. Wu, Attention-based convolutional neural networks for sentence classification, in: Proceedings of 17th Annual Conference of the International Speech Communication Association (INTERSPEECH), San Francisco, California, USA, 2016, pp. 705–709..
[37] K. Xu, J. Ba, R. Kiros, K. Cho, A. C. Courville, R. Salakhutdinov, R. S. Zemel, Y. Bengio, Show, attend and tell: Neural image caption generation with visual attention, in: Proceedings of the 32nd International Conference on Machine Learning (ICML), Lille, France, 2015, pp. 2048–2057..
[38] Z. Yang, D. Yang, C. Dyer, X. He, A. J. Smola, E. H. Hovy, Hierarchical attention networks for document classification, in: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, 2016, pp. 1480–1489..
[39] D. Bahdanau, K. Cho, Y. Bengio, Neural machine translation by jointly learning to align and translate, in: Proceedings of the 3rd International Conference on Learning Representations, San Diego, CA, USA, 2015..
[40] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, u. Kaiser, I. Polosukhin, Attention is all you need, in: Proceedings of the 31st International Conference on Neural Information Processing Systems, NIPS'17, Curran Associates Inc., Red Hook, NY, USA, 2017, p. 6000–6010..
[41] X. Zhang, J. J. Zhao, Y. Lecun, Character-level convolutional networks for text classification, in: Proceedings of the 29th Annual Conference on Neural Information Processing Systems, Montreal, Canada, 2015, pp. 649–657..
[42] T. Davidson, D. Warmsley, M. W. Macy, I. Weber, Automated hate speech detection and the problem of offensive language, in: International Conference on Weblogs and Social Media, 2017, 512–515..
[43] A.L. Maas, R.E. Daly, P.T. Pham, D. Huang, A.Y. Ng, C. Potts, Learning word vectors for sentiment analysis, in: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, Portland, Oregon, USA, 2011, pp. 142–150..
[44] C. Zhou, C. Sun, Z. Liu, F.C.M. Lau, A C-LSTM neural network for text classification, CoRR abs/1511.08630. arXiv:1511.08630, doi:1511.08630..
[45] S. Lai, L. Xu, K. Liu, J. Zhao, Recurrent convolutional neural networks for text classification, in: Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, AAAI Press, 2015, pp. 2267–2273..
[46] A. Joulin, E. Grave, P. Bojanowski, T. Mikolov, Bag of tricks for efficient text classification, in: Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, Association for Computational Linguistics, 2017, pp. 427–431..
[47] Z. Lin, M. Feng, C. N. dos Santos, M. Yu, B. Xiang, B. Zhou, Y. Bengio, A structured self-attentive sentence embedding, in: 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24–26, 2017, Conference Track Proceedings, 2017..
[48] R. Johnson, T. Zhang, Deep pyramid convolutional neural networks for text categorization, in: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Vancouver, Canada, 2017, pp. 562–570..