Pattern Recognition Letters 125 (2019) 264–270

Contents lists available at ScienceDirect

Pattern Recognition Letters

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

Fuzzy commonsense reasoning for multimodal sentiment analysis ☆

Iti Chaturvedi, Ranjan Satapathy, Sandro Cavallari, Erik Cambria*

School of Computer Science and Engineering, Nanyang Technological University Singapore

ARTICLE INFO

Article history:
Received 28 November 2018
Available online 30 April 2019

MSC:
41A05
41A10
65D05
65D17

Keywords:
Sentiment prediction
Fuzzy logic
Deep learning
Multi-modal

ABSTRACT

The majority of user-generated content posted online is in the form of text, images and videos but also physiological signals in games. AffectiveSpace is a vector space of affective commonsense available for English text but not for other languages nor other modalities such as electrocardiogram signals. We overcome this limitation by using deep learning to extract features from each modality and then projecting them to a common AffectiveSpace that has been clustered into different emotions. Because, in the real world, individuals tend to have partial or mixed sentiments about an opinion target, we use a fuzzy logic classifier to predict the degree of a particular emotion in AffectiveSpace. The combined model of deep convolutional neural networks and fuzzy logic is termed Convolutional Fuzzy Sentiment Classifier. Lastly, because the computational complexity of a fuzzy classifier is exponential with respect to the number of features, we project features to a four dimensional emotion space in order to speed up the classification performance.

1. Introduction

Sentiment analysis aims to classify text and video [1] into either positive, negative or neutral [2]. Detecting sentiments in social media such as text and video can help us understand opinions about products and events [3,4]. Recently, physiological signals such as electrocardiogram (ECG) are being used to understand personality and the effects of social interactions during game play [5]. Furthermore, media may be in multiple languages such as English or Spanish. Fusion of multimodal content has several challenges such as hyper-parameter tuning, interpretability, and speed.

In [6], the authors used unsupervised topic modeling for weighting the features from one modality before transferring to another modality. However, topic modeling is not suitable for predicting sentiments as it is unable to model underlying fine-grained sentiments such as 'frustration' and 'remorse'. Instead, in this paper we use deep learning [7] to extract features from each modality and then project them to AffectiveSpace [8], a vector space model of commonsense concepts such as 'beautiful painting' or 'poor writing' [9].

The complete network where nodes are concepts and edges determine the hierarchy among them is called SenticNet [10]. A dimensionality reduction of SenticNet results in AffectiveSpace [11]. It allows us to find semantics (semantically related concepts) for each given concept. Lastly, 24 basic emotions in the Hourglass model [12] are used as centroids to cluster AffectiveSpace. Fig. 1 shows a 2D view of the Hourglass model that classifies Level-1 emotions into 4 different categories. There are 4 Level-2 emotions formed by the composition of any two Level-1 emotions.

In the real world, individuals have partial or mixed sentiments about an opinion target, e.g., "iPhoneX has a nice touch screen, but it's very costly". A fuzzy logic classifier has membership functions that can range between partial positive and partial negative. Hence, it is ideal for modeling emotions in AffectiveSpace. Such features are easily able to adapt to the context in a particular domain (Books or Electronics) or a particular data type (Video or Text or Heart Signals) or a language (Spanish or English). In this way, we can solve multi-domain and multimodal challenges. Furthermore, such a model only requires two membership functions to combine the emotions resulting in low computational complexity.

The organization of the paper is as follows: Section 2 reviews related works and datasets on sentiment detection; Section 3 provides the preliminary concepts necessary to understand the present work; Section 4 details the proposed model for predicting sentiments in video and text; finally, in Section 5 we validate our method on different datasets and provide conclusions in Section 6.

2. Related work

Sentiment prediction aims to classify customer experiences as positive or negative. Emotion recognition is a fine-grained model accounting for each sentiment such as 'angry' or 'happy' [13]. Traditional methods for multimodal fusion of sentiments are unable

☆ **Conflict of interest:** There is no conflict of interest.

* Corresponding author.
E-mail address: cambria@ntu.edu.sg (E. Cambria).

https://doi.org/10.1016/j.patrec.2019.04.024
0167-8655/

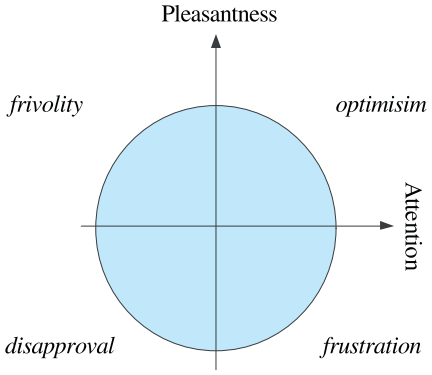

Fig. 1. Different values of Level 1 emotions result in Level 2 emotions.

to cope with the diversity of data types such as 'Text', 'Audio' or 'Video'. Furthermore, it becomes difficult to optimize parameters across different modalities and sub emotions simultaneously. For example, in [14] the authors employ multiple kernel learning for fusion of audio, visual and textual features that are extracted using deep learning.

A limitation of their method is large number of features that are difficult to visualize. Instead, we show that we can project both text and video features into a 4 dimensional AffectiveSpace and the complex emotions can be visualized by fuzzy blending of partial sub-emotions. In [15], the authors consider multi-lingual sentiment analysis by combining lexical features such as part of speech and word vectors. However, their method is unable to model emotions in video or signals. Another author used deep memory networks to capture the sentiment of individual words in restaurant reviews [16]. They consider an additional 'attention' node in each layer to capture the location of words. This method has quadratic complexity, hence we extract convolutional features using a sliding window to capture the context of words in relation to its neighbors [17].

Long short-term memory (LSTM) networks have also been used for summarizing sentiments in product reviews where each hidden neuron represents a single word. The LSTM model can remember long-range dependencies in a sentence such as between the first and the last word [18]. However, it is difficult to train LSTM as there are a large number of parameters. In contrast, we propose to use a recurrent neural network (RNN) model with a single memory state. Here, the input to RNN are bi-grams and tri-grams extracted using deep learning. In this way, we can reduce the dimensionality of the model.

Fig. 2 illustrates the deep convolutional model. The input is a sequence of individual words in a sentence. In the second layer, two consecutive words are combined to form significant bi-grams

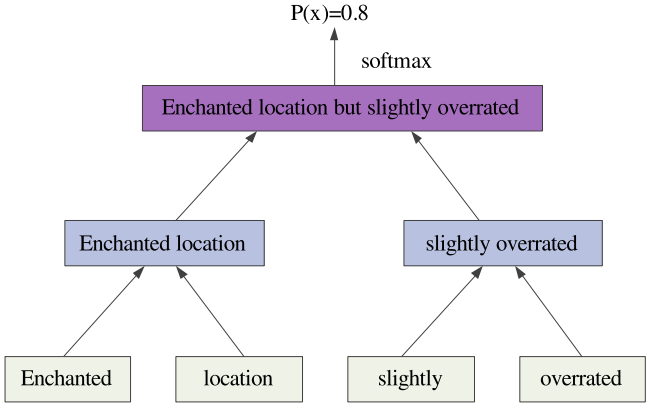

Fig. 2. Layers in a Deep Sentiment Model.

such as 'Enchanted Location'. Such bi-grams are combined in the third layer using logical connectors such as 'but'. In summary, each dimension of AffectiveSpace corresponds to a specific emotion such as 'Happy' or 'Sad'. Such annotations do not exist in languages such as 'Spanish'. The problem is amplified for the case of video where only facial expressions are used to determine sentiment or for physiological signals such as ECG. In this paper, we projected the features in videos and Spanish text to AffectiveSpace. Such features are easily able to adapt to the context in a particular 'domain' (Books or Electronics) or a particular data type such as (Video or Text or Signals). For video or ECG dataset, the input is the sequence of images in a video and the label is the polarity of facial expression. It will project the learned emotion features onto AffectiveSpace of emotions. Furthermore, fuzzy membership functions can be used to model complex partial emotions.

3. Preliminaries

In this section, we describe deep learning for classifying sentiments in sentences or images. Next, we propose the equation for temporal dependence between consecutive sentences or images. These features are then combined in the next section using fuzzy logic.

3.1. Unified input

Given a training sample (video, text or signal) and the corresponding sentiment label $y \in \{Positive, Neutral, Negative\}$, we transform the input into a 2D image representation with dimensions $L_x \times L_y$ as follows:

- Text: For sentences of maximum length L we represent each word with a corresponding pre-trained vector representation of dimension d that has been computed from co-occurrence data. When we concatenate the word vectors of all words in a sentence it results in a 2D input vector of dimension $L \times d$. Neural networks require that all input sentences are of equal length. We use padding with zeros to make them all of equal length. Since we use sliding window to extract bi-gram and tri-gram features, it is robust to variations in sentence length. Hence, the padded zeros are simply discarded during convolution.
- Video: We convert videos into a sequence of images at a frequency of 50 images per minute. Next, we crop the images using the face boundaries resulting in a 2D input of dimension $L_x \times L_y$. The height and width of faces varies across the samples. Hence, we used padding with zeros in order to make them of equal dimension and also maintain the height/width ratio of the face. This will also ensure that the height/width ratio across different faces is not altered.
- ECG: For the ECG signal we consider the d lead signals (collected at different body parts) as the feature vectors at each time point. We consider 512 samples spanning 3 heart beats resulting in an input of dimension $512 \times d$.

3.2. Restricted Boltzmann Machines

A Restricted Boltzmann Machine (RBM) [19] is a neural model consisting of two layers (known as visible and hidden layers). RBM is trained in an unsupervised way to learn the joint probability distribution of the dataset. The state $\hat{h}_j$ of the hidden neuron j, with bias b_j, is a weighted sum over all continuous visible nodes $\boldsymbol{v}$ and is given by:

$$h_j = \frac{1}{1 + e^{-\hat{h}_j}} \text{ and } \hat{h}_j = b_j + \sum_i v_i w_{ij} \quad (1)$$

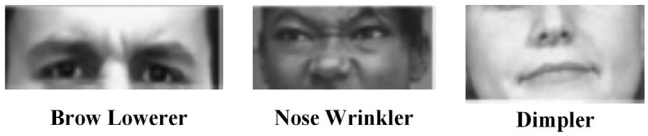

Fig. 3. Features in 'Happy' and 'Sad' faces.

where w_{ij} is the connection weight to hidden neuron j from visible node v_i. The binary state h_j of the hidden neuron can be defined by a sigmoid activation function. Similarly, in the next iteration, the binary state of each visible node v_i is reconstructed.

Lastly, the weights w_{ij} are updated as the difference between the original data and reconstructed visible layer labeled as the vector $\boldsymbol{v}_{recon} = (v_1, v_2, \ldots, v_n)_{recon}$ for n neurons, using:

$$\triangle w_{ij} = \alpha(< v_i h_j >_{data} - < v_i h_j >_{recon}) \tag{2}$$

where α is the learning rate and $< v_i h_j >$ is the expected frequency with which visible unit i and hidden unit j are active together when the visible vectors are sampled from the training set and the hidden units are determined by Eq. (1) [20].

3.3. Convolutional deep belief network

The modeling power of traditional RBM remained quite limited until it was proposed to stack many RBM in a hierarchical manner giving the rise to the well-known deep belief network (DBN) model. As proposed in [21] it is possible to create a Convolutional RBM (CRBM) by naturally extending a traditional RBM in 2 dimensions using the convolution operation. Similarly, if we stack many layers of CRBMs, it is possible to create to a Convolutional Deep Belief Network (CDBN), where we simply partition the hidden layer into Z groups.

Each of the Z groups is associated with a $n_x \times n_y$ kernel where n_x is the width of the kernel and n_y is the height of the kernel. It is useful to note that for text or signal input the $n_y = d$ is fixed since it is impossible to split features of a single word or heart sample. Fig. 3 illustrates three features learned by CDBN for sentiment prediction. For example, a 'nose wrinkler' would correspond to 'angry' emotion and a 'dimpler' would indicate 'happy' emotion. Let us assume that the input has dimension $\mathrm{L_x} \times \mathrm{L_y}$. Then, the convolution will result in a hidden layer of Z groups each of dimension $(\mathrm{L_x} - n_x + 1) \times (\mathrm{L_y} - n_y + 1)$. These learned kernel weights are shared among all hidden units in a particular group. To train such a multi-layer system, we must compute the gradient of the total energy function E with respect to the weights in all the layers. The energy function of layer l is now a sum over the energy of individual blocks given by:

$$\mathrm{E}^l = -\sum_{z=1}^{\mathrm{Z}} \sum_{\substack{i,j \\ (\mathrm{L_y}-n_y+1)}}^{\substack{(\mathrm{L_x}-n_x+1), \\ (\mathrm{L_y}-n_y+1)}} \sum_{r,s}^{n_x,n_y} v_{i+r-1,j+s-1} h_{ij}^z w_{rs}^l. \tag{3}$$

In order to obtain the desired number of output features and to flatten out all convolution filters, a softmax logistic layer is introduced as last layer of the CDBN as shown in Fig. 2.

3.4. Recurrent neural networks

RNNs have temporal memory and hence are ideal to model a sequence of sentences [14]. The CDBN is not able to capture the causality between k-grams in consecutive sentences or images $s(t)$ and $s(t+1)$. To overcome this limitation, we added a recurrent layer of neurons which takes as input the n_h features extracted by the last logistic layer of the CDBN. The standard RNN output, $\mathbf{x}_l(t)$, at time step t for each layer l is calculated using the following equations:

$$\mathbf{x}_l(t) = f(W_{\mathrm{R}}^l.\mathbf{x}_l(t-1) + W_l.\mathbf{x}_{l-1}(t)) \tag{4}$$

where W_{R} is the interconnection matrix among hidden neurons and W_l is the weight matrix of connections between hidden neurons and the input nodes, $\mathbf{x}_{l-1}(t)$ is the input vector at time step t from layer $l-1$, vectors $\mathbf{x}_l(t)$ and $\mathbf{x}_l(t-1)$ represent hidden neuron activation at time steps t and $t-1$, respectively, and f is the non-linear activation function. RNN are trained using standard back propagation through time algorithm and learns a compact representation of n_r features.

4. Convolutional fuzzy sentiment classifier

In this section, we first describe fuzzy membership functions for modeling partial emotions in sentences. Next, we describe our proposed framework for integrating emotions into the features learned by CDBN. The resulting model is termed Convolutional Fuzzy Sentiment Classifier (CSFC). We also show that the computational complexity of the combined model is much lower than the traditional fuzzy classifier.

4.1. Fuzzy sentiment classifier

In this paper, we consider the variations in emotions for predicting the sentiment in a sentence. Each emotion in AffectiveSpace can be divided into 6 sub-emotions. The emotions follow a normal distribution $\mathcal{N}\sim(0,1)$. The input is four dimensional corresponding to four emotions. The fuzzy classifier predicts positive, neutral and negative sentiments using fuzzy memberships over these input emotions.

The four emotional dimensions for sentence $s(t)$ are, the pleasantness $m_p(t)$, the sensitivity $m_s(t)$, the attention $m_a(t)$, and the aptitude $m_d(t)$ have uncertainties, which vary in a given range, i.e., $m_p(t) \in [m_{\mathrm{pmin}}, m_{\mathrm{pmax}}]$, $m_s(t) \in [m_{\mathrm{smin}}, m_{\mathrm{smax}}]$, $m_a(t) \in [m_{\mathrm{amin}}, m_{\mathrm{amax}}]$ and $m_d(t) \in [m_{\mathrm{dmin}}, m_{\mathrm{dmax}}]$. It is to say that the uncertainty of the pleasantness $m_p(t)$ is bounded by its minimum value m_{pmin} and its maximum value m_{pmax}. Similarly, the other affective dimensions are bounded by their minimum and maximum values. Here, we set the input range for each emotion between $G(1) = 0.24$ and $-G(1) = -0.24$ where $G\sim\mathcal{N}(0,1)$. This range can be divided into sub-emotions. For example, if the value for Pleasantness dimension is between $G(2/3)$ and $G(1/3)$ then the emotion is 'Joy' [8].

For illustration purpose, let us consider two affective dimensions pleasantness $m_p(t)$, and attention $m_a(t)$ as defined above. Consider the attention level for a sentence $s(t)$:

$$y_a(t) = m_a s(t) \tag{5}$$

where m_a represent the attention level in the sentence $s(t)$, $\widetilde{m_a} \leq m_a \leq \widehat{m_a}$, where $\widetilde{m_a}$ and $\widehat{m_a}$ are the constant scalars and are used to constrain lower and upper bounds of the attention level. The following three cases are considered corresponding to three different sentiment conditions:

1. $\widetilde{m_a} = \widehat{m_a} = 0$; then $m_a = 0$, which implies that the corresponding sentiment is completely negative.
2. $\widetilde{m_a} = \widehat{m_a} = 1$; then $m_a = 1$, which implies that the corresponding sentiment is completely positive.
3. $0 < \widetilde{m_a} < \widehat{m_a} < 1$, which means that their exists a weak sentiment in the corresponding sentence or it could be a neutral or factual sentence.

Next, we obtain the values of $1/m_p(t)$ and $1/m_a(t)$ as given in Eq. (8). We have only provided the equations for attention dimension and membership functions M_1 and M_2. The corresponding equations for pleasantness dimensions and membership functions

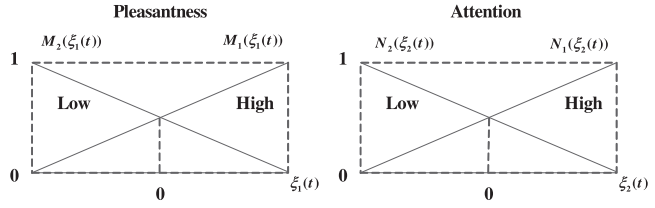

Fig. 4. Membership functions for Pleasantness and Attention.

N_1 and N_2 can be computed in the same way. If we assume that membership functions $M_1(\xi_1(t))$, $M_2(\xi_1(t))$, $N_1(\xi_2(t))$ and $N_2(\xi_2(t))$ such that:

$$M_1(\xi_1(t)) + M_2(\xi_1(t)) = 1 \qquad (6)$$
$$N_1(\xi_2(t)) + N_2(\xi_2(t)) = 1$$

where $\xi_1(t) = 1/m_p$ and $\xi_2(t) = 1/m_a$. Then, we can get Eq. (8).

The range of membership functions are labeled 'High'(m_{pmax}) and 'Low'(m_{pmin}) as shown in Fig. 4, where 'High' corresponds to positive sentiment such as 'Joy' and 'Low' corresponds to negative sentiments such as 'Anger'. When the membership is near 0 then the emotion will be neutral or mixed such as in 'Surprise'. The partial membership to both the functions M_1 and M_2 such as 'Very Low, Low, High, Very High' can be determined using Eq. (8). For example, when $1/m_p(t) = \widehat{m_p}$ is maximum as shown in Eq. (8) then $M_1 = 1$ and $M_2 = 0$. Lastly, fuzzy blending allows us to infer the overall fuzzy sentiment model given by Eq. (7) using rules where $K_{i=1:4}$ are the weight matrices to be learned during training. For illustration, we have only shown Rule 1 where $\xi_1(t)$ is High and $\xi_2(t)$ is Low. Similarly, we can create four rules for two discrete levels of the two emotions. Eq. (8) describes Rule 1 as an example. Each weight matrix K_i is of dimension n^*m where n is the number of inputs (= 4) and m is the number of outputs (= 2). These matrices are determined so that the fuzzy neural network is stable over time. Similar to a neural network the final output of a fuzzy neural network is now a summation over the activation of all the rules as follows:

$$y(t) = \sum_{i=1}^{4} K_i s(t) \qquad (7)$$

4.2. Convolutional fuzzy sentiment framework

In this section, we explain the complete framework for creating fuzzy membership functions for sentiment prediction using CDBN. We first construct a minimal CDBN with visible layer of $L \times d$ nodes, where L is length of the sentence and d is the word vector length; there are several hidden convolution layers of k-gram neurons, then there is a penultimate hidden logistic layer of n_h neurons and the last layer is output neurons each class $n_d \in \{+, 0, -\}$ where ' + ' is positive, '0' is neutral and ' - ' is negative review. The n_h features learned in the penultimate logistic layer of CDBN after training is used as the new low dimensional input data to the RNN.

$$\max \frac{1}{m_p(t)} = \frac{1}{m_{pmin}(t)} = \widehat{m_p},\ \min \frac{1}{m_p(t)} = \frac{1}{m_{pmax}(t)}$$
$$= \widetilde{m_p}, M_1(\xi_1(t)) = \frac{\frac{1}{m_p(t)} - \widetilde{m_p}}{\widehat{m_p} - \widetilde{m_p}},\ M_2(\xi_1(t))$$
$$= \frac{\widehat{m_p} - \frac{1}{m_p(t)}}{\widehat{m_p} - \widetilde{m_p}}$$

Rule 1: IF $\xi_1(t)$ is High and $\xi_2(t)$ is Low,THEN $y(t) = K_1 s(t)$ (8)

Next, we construct a RNN with n_h input nodes and n_r hidden neurons with time-delays. The n_r features expressed at the hidden neurons after training: from the new input data of T samples. Lastly, we project the n_r features to four dimensional AffectiveSpace using:

$$\mathbf{s}_{T,4} = \mathbf{s}_{T,n_r} \times A_{1:n_r,4} \qquad (9)$$

where A is AffectiveSpace and $\mathbf{s} = (s(1), s(2), \ldots, s(T))$ is the vector of sentence features. Since AffectiveSpace has 100,000 concepts (both single words and multi-word expressions [22]), we perform a dimensionality reduction to n_r concepts before projection. Each test sample is then used to generate an embedding of dimension n_h features from CDBN and then n_r features from RNN and finally classified by the fuzzy classifier.

To determine the number of hidden layers in the CDBN, we compute the change in visible layer reconstruction error $\triangle\epsilon$ on the training samples. This is the root mean square error between input training sample and reconstructed sample at each visible node. If there is a significant change in the error $\triangle\epsilon$, a new hidden layer is added. The above progresses iteratively until additional hidden layers do not change the classification precision error significantly, and the optimal configuration is achieved. Following [23], to determine the optimal number of hidden neurons in a single layer, we consider the number of components with eigenvalues above a threshold after during principle component analysis. The contrastive divergence approach will sample features with high frequency into the upper layers, resulting in the formation of k-grams at hidden neurons.

We first construct a minimal deep CNN with visible layer of $L_x \times L_y$. The sentence model is a simple extension where L_x is the length of the sentence L and L_y is the word vector length d. The first hidden convolution layer of learns features of size $n_x \times n_y$, second hidden logistic layer of n_h neurons and n_d output neurons. For sentences, we simply substitute $n_x = k$ and $n_x = d$ to obtain k-gram word features. The n_h features expressed at logistic layer after training form the new input data of T samples. Next, we construct a RNN with n_h input nodes and n_r hidden neurons with time-delays. The n_r features expressed at the hidden neurons after training form the new input data of T samples. Lastly, we train a fuzzy classifier with n_r features and T samples. Each test sample is used to generate n_h outputs from deep CNN and n_r outputs from RNN and finally classified using fuzzy logic. Fig. 5 illustrates

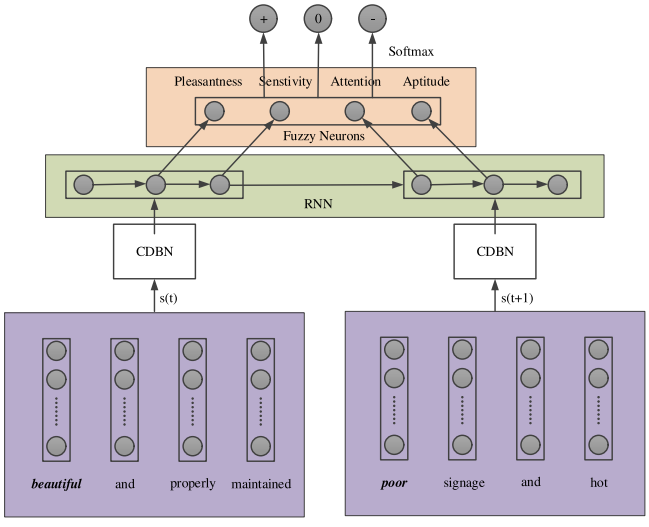

Fig. 5. State diagram of Convolutional Fuzzy Sentiment Classifier.

Table 1
Comparison of different baselines with proposed CFSC on different types of Dataset.

Type	Dataset	Baseline (B)	Acc (B)	CRNN	CFSC
Text	Alh	CDBN [14]	76.1	80.2	**91.3**
Text	Mos	CDBN [14]	86.3	94.4	**99.1**
Text	Sag	CDBN [14]	82.3	88.2	**95.7**
Video	MOUD	SVM [26]	67.2	94.1	**97.4**
ECG	Ami	NB [5]	53.1	55.2	**59.6**
CrossD	B → D	R3 [27]	71.1	88.4	**93.5**
CrossD	B → D	TDN [28]	85.3	88.4	**93.5**
CrossD	B → D	R3 [27]	71.1	88.4	**93.5**

the complete framework where features in each pair of sentences at time t and $t+1$ are combined.

4.3. Computational complexity

The fuzzy classifier complexity is $O(\sum_{i=1}^{n_i} 2n_o + \sum_{i=1}^{n_i} k)$ [24], where n_i and n_o are the number of input and output feature maps and k is the number of membership functions for each input feature. It is easy to see that a fuzzy classifier becomes very slow with the number of input dimensions and the number of membership function. Hence, in this paper we project the features learned by CDBN to a 4 dimensional input space. Furthermore, we design 2 membership functions to model partial emotions where upper and lower limit correspond to positive and negative respectively. In this way, fuzzy blending can be used to create rules for complex emotions. Hence, the cost of the proposed fuzzy classifier is constant $O(\sum_{i=1}^{4} 2n_o + \sum_{i=1}^{4} \times 2)$ where n_o is the number of output classes.

5. Experiments

Validation of the proposed CFSC method (available on GitHub[1]) is performed on four real-world benchmarks. Next, we discuss the parameter setting. Lastly, we visualize the 24 sub-emotions in the reviews. Following previous authors, we consider Accuracy[2] metric to evaluate the models. This allows us to directly compare with their results.

5.1. Trip advisor text dataset

In this section, we consider a dataset that was taken from TripAdvisor website. We scrap reviews on three well-known monuments in Spain: Alhambra (Alh: 9253), Mosque of Cordoba (Mos: 4619) and Sagrada Familia (Sag: 43566). The reviews have expert ratings from 1 to 5, where 1 is strongly negative, 2 is weakly negative, 3 is neutral, 4 is weakly positive and 5 is strongly positive.

Hence, we consider a 5 class problem and 10 fold cross-validation. Table 1 shows the comparison of accuracy with CDBN and CRNN (CDBN followed by RNN), we notice over 7% improvement for the Alhambra and Sagrada Familia dataset. The percentage improvement is highest for Sagrada Familia, hence the method works better as dataset size increases.

5.2. Youtube video review dataset

For our experiment, we use the multimodal Opinion Utterances Dataset (MOUD) dataset developed by Morency et al. [25]. They started collecting videos from popular social media (e.g., YouTube) using several keywords (e.g., "favorite products") to produce search results consisting of videos of either product reviews or recommendation. On average, each video has 6 utterances and each utterance is 5 seconds long. Each utterance in a video is annotated separately as positive or negative. Hence, sentiment can change during a product review.

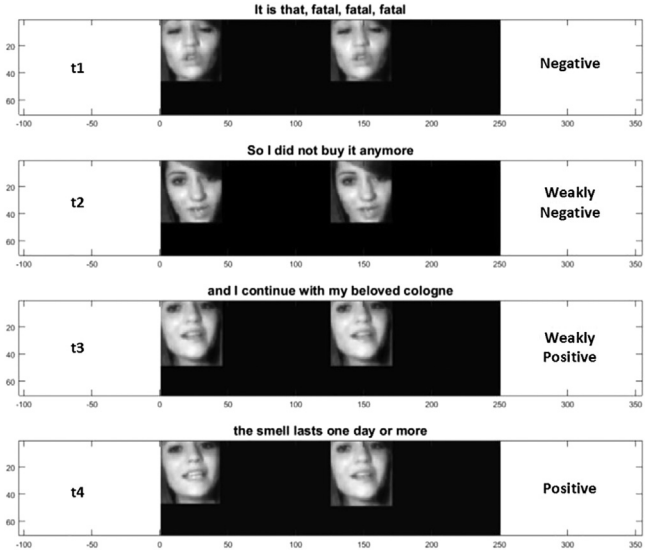

Fig. 6. Four consecutive video frames (t1 to t4) in a You-Tube video. To capture temporal dependence, we transform each pair of consecutive images at t and $t+1$ into a single image.

Following Pérez-Rosas et al. [26], we consider 448 utterances labeled positive or negative. To capture temporal dependence, we transform each pair of consecutive images at t and $t+1$ into a single image. Table 1 compares the accuracy of CFSC with baselines. We have a marginal improvement over CRNN. However, there is over 30% improvement over the support vector machine (SVM) classifier used in [26]. Furthermore, it is possible to visualize the test images using the values of different Affective emotions. Fig. 6 shows Affective emotions in a perfume review video sequence. Here, test images with low value of different emotions were called 'weakly positive' and images with extreme values of affective emotions were labeled 'strongly positive'. We can see that the review is starts as negative ($t1$ and $t2$) and then becomes positive at $t3$ and $t4$.

5.3. Physiological signals ECG dataset

Amigos (Ami) database contains ECG recordings from 40 subjects and 16 movie clip [5]. Each clip targeted one of the following nine emotions: amusement, excitement, happiness, calmness, anger, disgust, fear, sadness, and surprise. To avoid contaminating data recordings with multiple emotions, only the recordings captured during the last 60 s of each film clip were used for analysis. A 5 s baseline recording showing a fixation cross was shown before each film clip in order to help the subject return to a neutral emotional state.

Each participant performed an initial self-assessment for valence ranging from 1 (unpleasant/stressed) to 9 (happy/elated) (see Fig. 7). We consider 2 leads and up to 500 samples from each lead and binary valence labels. In Table 1 we compare the F1 score of CFSC with baselines. We have a marginal improvement over CRNN. However, there is over 6% improvement over the Naive Bayes (NB) classifier used in [5].

[1] http://github.com/senticnet/convolutional-fuzzy-classifier .
[2] (tp+tn)/(tp+fp+tn+fn)

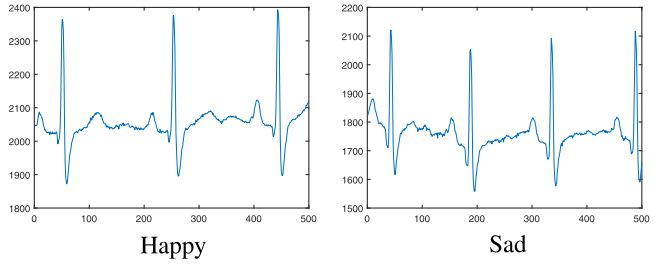

Fig. 7. Sample ECG when watching 'Happy' and 'Sad' movie clips. The horizontal axis in the heart-beat time stamp and the vertical axis is the electrical activity of the heart.

5.4. Cross domain dataset

Lastly, we verify the effectiveness of CFSC in classifying subjective sentences using the multi-domain sentiment analysis dataset [29]. Following previous authors, we first report the results on the binary problem of classifying reviews as positive (4 or 5) and negative (1 or 2). The four domains consist of 'Books' (B), 'DVD' (D), 'Electronics' (E), and 'Kitchen' (K) reviews, where each domain contains 2000 reviews. Hence, as an illustration training data in the form of 1000 positive and 1000 negative reviews were taken.

We construct a cross-domain (CrossD) task of sentiment classification on this dataset. Here, 2000 reviews in one domain are the training data and 2000 reviews in a different domain are the test data. Table 1 shows the comparison for different methods. The proposed CFSC outperforms Transfer Deep Network (TDN) [28] (85%) by over 5% and R3 [27] (71%) by over 10% in accuracy. In TDN [28], the authors considered two parallel deep auto-encoders to learn transferable features and classification features. However, they do not use convolutional neural networks; hence, they are unable to capture the context of words. In Rule3(R3) [27], the authors proposed three rules that must be satisfied for cross-domain classification. They considered handcrafted features, instead in our method we automatically learn cross-domain features. We also compared the performance of different modules in CFSC. Next, we consider the performance of only CDBN (71%) and CRNN (88%). Hence, the fuzzy classifier results in over 5% improvement.

5.5. Parameter setting

We have used pre-trained word vectors for English. Following previous authors, the word vector length was empirically set to 300, and unknown words were randomly initialized [30]. To determine the number of hidden neurons and layers we consider the mean square classification error on training data. We see a significant improvement as we use up to 5 hidden layers. This is possible because each layer is trained independently of the layer below, thus there are a small number of parameters in each layer. With increasing number of parameters during training the model is not able to generalize to unseen data.

Here, the model over-fits to the training data and the accuracy on new test data is low. If we train each layer independently, then the complexity of the model is low and hence over-fitting is avoided. Since fuzzy classifier is able to model the uncertainties in the data, extensive parameter tuning was not required. Each simulation takes about 10 min to complete. Our best results are obtained with an ensemble of CFSC 10-fold cross-validation that differ in their random initialization and mini-batches of 100 samples.

In Fig. 8, we can see that the model is robust to small changes in parameters for predicted Emotions in 'Books' to 'DvD' classifier. To measure the stability of the model to parameters we consider the reduction in error with each iteration of training. For example, in Fig. 8(a) when we increase the number of neurons in all the hidden layers form 10 to 15 the error curve does not change significantly. Similarly, in Fig. 8(b), if we change the learning rate from 0.01 to 0.1 there is only a slight change in the error curve. Lastly, we can see that our model with two membership functions (MF) is exponentially faster than a model with three or more MF's.

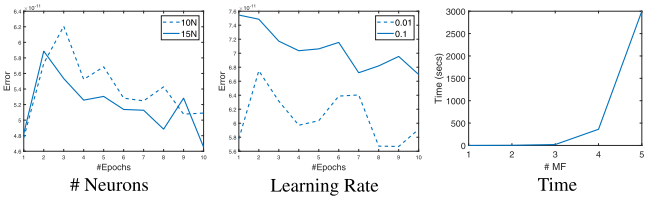

Fig. 8. Plot for Error Vs # Iterations for (a) Different number of neurons (b) Different learning rates (c) Time (secs) Vs # MF's.

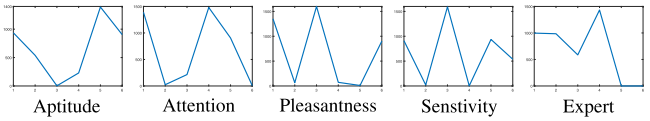

Fig. 9. Distribution of Sub-emotions in AffectiveSpace for 'Books' and 'DvD' reviews. The expert rating distribution is also shown. The horizontal axis in the 6 sub-emotions and the vertical axis is the frequency of reviews in each sub-emotion.

5.6. Visualization of fuzzy emotions

Fig. 9 shows the distribution of predicted Emotions in 'Books' and 'DvD' reviews using a fuzzy classifier. For each emotion in AffectiveSpace, we plot the number of reviews in each of the six sub-emotions define in [8].

The expert rating distribution has the highest number of reviews with rating 4 (strongly positive). However, if we look at the highest frequency of emotions, for Pleasantness and Sensitivity the emotions are slightly negative and for Attention and Aptitude they are only slightly positive. Hence, by looking at the sub-emotions, we can understand the dataset better and the expert rating may in fact be incorrect in judging the true emotions of users.

6. Conclusion

In this paper, we have proposed a sentiment classifier based on emotions that is able to predict sentiments in video and text sequences. Our simulation and experimental study showed that the proposed CFSC outperformed the baseline methods in classification accuracy. The method was also able to visualize 24 different emotions in test samples. We observed an improvement in the range of 10–20% in accuracy.

Unlike traditional LSTM, which models every word with a single neuron, in this paper we use a low-dimensional RNN to classify concepts learned via deep learning. By projecting temporal features onto AffectiveSpace, we are able to interpret the features learned. Lastly, we take into account that most sentences have mixed emotions such as sarcasm that can only be modeled effectively using fuzzy membership functions. Hence, we predict the final accuracy of the classifier using fuzzy blending over each pair of simple emotions.

Acknowledgement

This work is partially supported by the Data Science and Artificial Intelligence Center (DSAIR) at the Nanyang Technological University.

References

[1] E. Cambria, D. Rajagopal, D. Olsher, D. Das, Big Social Data Analysis, in: R. Akerkar (Ed.), Big Data Computing, Chapman and Hall/CRC, 2013, pp. 401–414.

[2] I. Chaturvedi, E. Ragusa, P. Gastaldo, R. Zunino, E. Cambria, Bayesian network based extreme learning machine for subjectivity detection, J. Franklin Inst. 355 (4) (2018) 1780–1797.

[3] K. Cheng, J. Li, J. Tang, H. Liu, Unsupervised sentiment analysis with signed social networks, AAAI, 2017.

[4] N. Howard, E. Cambria, Intention awareness: improving upon situation awareness in human-centric environments, Human-centric Comput. Inf. Sci. 3 (9) (2013).

[5] J. Abdon Miranda-Correa, M. Khomami Abadi, N. Sebe, I. Patras, Amigos: A dataset for mood, personality and affect research on individuals and groups (2017).

[6] X. Huang, Y. Rao, H. Xie, T.-L. Wong, F.L. Wang, Cross-domain sentiment classification via topic-related tradaboost, AAAI, 2017.

[7] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, in: NIPS, Curran Associates, Inc., 2012, pp. 1097–1105.

[8] E. Cambria, J. Fu, F. Bisio, S. Poria, Affectivespace 2: enabling affective intuition for concept-level sentiment analysis, in: AAAI, 2015, pp. 508–514.

[9] E. Cambria, A. Hussain, C. Havasi, C. Eckl, Common sense computing: from the society of mind to digital intuition and beyond, in: J. Fierrez, J. Ortega, A. Esposito, A. Drygajlo, M. Faundez-Zanuy (Eds.), Biometric ID Management and Multimodal Communication, Lecture Notes in Computer Science, 5707, Springer, Berlin Heidelberg, 2009, pp. 252–259.

[10] E. Cambria, S. Poria, D. Hazarika, K. Kwok, SenticNet 5: discovering conceptual primitives for sentiment analysis by means of context embeddings, in: AAAI, 2018, pp. 1795–1802.

[11] E. Cambria, A. Hussain, C. Havasi, C. Eckl, Sentic computing: exploitation of common sense for the development of emotion-sensitive systems, in: A. Esposito, N. Campbell, C. Vogel, A. Hussain, A. Nijholt (Eds.), Development of Multimodal Interfaces: Active Listening and Synchrony, Springer, Berlin, 2010, pp. 148–156.

[12] E. Cambria, A. Livingstone, A. Hussain, The hourglass of emotions, in: A. Esposito, A. Vinciarelli, R. Hoffmann, V. Muller (Eds.), Cognitive Behavioral Systems, Lecture Notes in Computer Science, 7403, Springer, Berlin Heidelberg, 2012, pp. 144–157.

[13] E. Cambria, A. Hussain, C. Havasi, C. Eckl, SenticSpace: visualizing opinions and sentiments in a multi-dimensional vector space, in: R. Setchi, I. Jordanov, R. Howlett, L. Jain (Eds.), Knowledge-Based and Intelligent Information and Engineering Systems, Lecture Notes in Artificial Intelligence, 6279, Springer, Berlin, 2010, pp. 385–393.

[14] S. Poria, I. Chaturvedi, E. Cambria, A. Hussain, Convolutional MKL based multimodal emotion recognition and sentiment analysis, in: ICDM, Barcelona, 2016, pp. 439–448.

[15] M. Giatsoglou, M.G. Vozalis, K. Diamantaras, A. Vakali, G. Sarigiannidis, K.C. Chatzisavvas, Sentiment analysis leveraging emotions and word embeddings, Expert Syst. Appl. 69 (2017) 214–224.

[16] D. Tang, B. Qin, T. Liu, Aspect level sentiment classification with deep memory network, in: EMNLP, 2016, pp. 214–224.

[17] M. Yang, W. Tu, J. Wang, F. Xu, X. Chen, Attention based LSTM for target dependent sentiment classification, AAAI, 2017.

[18] J. Xu, D. Chen, X. Qiu, X. Huang, Cached long short-term memory neural networks for document-level sentiment classification, in: EMNLP, 2016, pp. 1660–1669.

[19] A. Fischer, C. Igel, An introduction to restricted Boltzmann machines, in: ICPR, Springer, 2012, pp. 14–36.

[20] I. Chaturvedi, Y.-S. Ong, I.W. Tsang, R.E. Welsch, E. Cambria, Learning word dependencies in text by means of a deep recurrent belief network, Knowl. Based Syst. 108 (2016) 144–154.

[21] H. Lee, R. Grosse, R. Ranganath, A.Y. Ng, Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations, in: Proceedings of the 26th annual international conference on machine learning, ACM, 2009, pp. 609–616.

[22] D. Rajagopal, E. Cambria, D. Olsher, K. Kwok, A graph-based approach to commonsense concept extraction and semantic similarity detection, in: WWW, 2013, pp. 565–570.

[23] M. Tanaka, M. Okutomi, A novel inference of a restricted Boltzmann machine, in: ICPR, 2014, pp. 1526–1531.

[24] P. Baranyi, K.-F. Lei, Y. Yam, Complexity reduction of singleton based neuro–fuzzy algorithm, in: IEEE SMC, 4, 2000, pp. 2503–2508 vol.4.

[25] L.-P. Morency, R. Mihalcea, P. Doshi, Towards multimodal sentiment analysis: Harvesting opinions from the web, in: ICMI, ACM, 2011, pp. 169–176.

[26] V. Pérez-Rosas, R. Mihalcea, L.-P. Morency, Utterance-level multimodal sentiment analysis, in: ACL, 2013, pp. 973–982.

[27] D. Bollegala, T. Mu, J.Y. Goulermas, Cross-domain sentiment classification using sentiment sensitive embeddings, IEEE Trans. Knowl. Data Eng. 28 (2) (2016) 398–410.

[28] M. Long, J. Wang, Y. Cao, J. Sun, P.S. Yu, Deep learning of transferable representation for scalable domain adaptation, IEEE TKDE 28 (8) (2016) 2027–2040.

[29] M. Dredze, K. Crammer, F. Pereira, Confidence-weighted linear classification, in: ICML, 2008, pp. 264–271.

[30] Y. Kim, Convolutional neural networks for sentence classification, in: EMNLP, 2014, pp. 1746–1751.