Emotion Classification and Its Application on Humanoid Robot

劉　寧

A Thesis submitted to Tokushima University in partial fulfillment of the requirements for the degree of Doctor of Philosophy

2018

Tokushima University
Graduate School of Advanced Technology and Science
Information Science and Intelligent Systems

Contents

List of Figures

List of Tables

Acknowledgment

This thesis represents not only my work at the keyboard, it is a four years' memory of living in Tokushima with my lovely friends, specially the A1 Group. I have to thank all those who have encouraged and helped me in my daily life during my first year in Japan. Above all, I must acknowledge my indebtedness to the following teachers and friends:

First and foremost, I wish to thank my advisor, professor Fuji Ren, director of the Tokushima University Ren Laboratory, for the continuous support of my Ph.D research and related exploration. With his patience, motivation, immense knowledge and great support, I can focus on affective computing and humanoid robot freely with special exotic culture experience of Japan. I was awarded a scholarship by CSC council under the supervision of Prof. Ren, which enabled me to pursue my study in Tokushima. I really appreciate that. My sincere thanks also goes to Dr. Shun Nishide and Dr. Xin Kang for their insightful discussion on my researches. I must thank Dr. Haitao Yu for helping with my newbie life after landing on Japan.

Dozens of people have helped and inspired me immensely at the Ren Lab. Thanks to Chao Li, Zhichao Cui, Xudong Zhang, Duo Feng and Mengjia He. A special acknowledgment goes to the members of CSSA at Tokushima, for our tacit teamworks on many activities.

I would thank MS. Chieko Nomura, the Japanese language class teacher of OASIS, who taught me elementary Japanese and helped me to understand the society and culture of Japanese. Specifically, I would thank Prof. Kenji Kita and Prof.Masami Shishibori, who had contributed so much time and efforts in reviewing this thesis. Their marvelous suggestions and insightful comments helped too much to improve this thesis.

Finally, I am deeply thanks to my girlfriend Sisi, who supports and encourages me to finish my Ph.D degree with tender care and endless love.

Abstract

In this thesis, the emotion classification based on Ren_CECps and its application on humanoid robot REN-XIN are proposed. "Affective Computing" provided by Picard in 1997, which is of great importance and is computing that relates to, arises from, or deliberately influences emotion or other affective phenomena. The recognition model in our research is trained from a multi-label Chinese emotion corpus annotated by Ren Lab(Ren_CECps). In order to find the complex distributions of every annotated sentences, we be the first group to make a 2D graph using t-SNE(t-Distributed Stochastic Neighbor Embedding) algorithm, and a 3D visualization map is also proposed. To avoid the point overlap, we propose an emotion separated $TF \cdot IDF$(SeTF $\cdot$ IDF) algorithm to assign one multi-label annotated sentence with different feature vectors for every single category.

The 2D and 3D reduced distribution maps gives us a clearer view of the distance within every separated emotion sentence. The points position changed between SeTF $\cdot$ IDF and $TF \cdot IDF$ inspired us to apply a distance measurement algorithm to recognize the emotion categories. We finally propose a fast WMD method which is a 16000 times faster version of Word Mover's Distance(WMD) algorithm. Utilizing the distance features generated by fast WMD method, our experiments show that the SVM classifier get the best F1 scores of 0.318 than the features calculated by SeTF $\cdot$ IDF and $TF \cdot IDF$ of F1 scores of 0.293 and 0.203 respectively. Our cross-language experiments based on Chinese emotional corpus Ren_CECps and English news dataset 20 newsgroup show that with the fast WMD computed features, SVM classifiers get 3 times and 9 times improvement of F1 scores respectively compared with the same dimension features reduced from tradition $TF \cdot IDF$.

Despite the huge progress achieved in robot field, the expression controls of humanoid robot with visual human-likeness face are still manually operated by developer for specified or limited scenarios. With the 'soul' embedded with the emotion recognition model trained with the distance features above, we try to enhance the 'body' of our humanoid robot REN-XIN by improve the expression ability. We utilized the proposed fast WMD method

which can recognize nine emotion categories in texts as emotional trigger to generate the corresponding action labels according to the robot's response. For the robot system, running the computed basic expression and the voice at the same time, we can get an acceptable humanoid robot interaction with emotion expression.

During the running interaction with Actroid REN-XIN, the fast WMD based emotional trigger system needs at least 7s to deal with the response. To make a real time interaction, the seamless user experience is a essential aspect. Thereby, for people are communicating with humanoid robot, the delayed feedback results to no long communication desire. To solve this rough gap, we propose a CNN+LSTM based DNN model. In the experiments, we utilize the same sub-data sets of the Chinese emotional corpus(Ren_CECps) used in fast WMD experiments. The experiments are proceeded in fast WMD, CNN+LSTM, CNN and LSTM respectively. The results show that CNN+LSTM gets the best result of F1 score 0.35 in 1v1 experiment, and almost the same accuracy with fast WMD of F1 scores 0.367 with 0.366 in 4v1 experiment. In the training process, our experiments show that the DNNs only need 3 epochs to finish training. This is not only the difference between minutes and weeks cost in training, but also the extended flexibility for the actroid robot. Our contributes show the CNN+LSTM model has excellent ability for emotion classification and robot control with time sensitive.

Chapter 1

Introduction

周穆王西巡狩，越昆仑，不至弇山。反还，未及中国，道有献工人名偃师，穆王荐之，问曰：“若有何能？”偃师曰：“臣唯命所试。然臣已有所造，愿王先观之。”穆王曰：“日以俱来，吾与若俱观之。”越日，偃师谒见王。王荐之曰：“若与偕来者何人邪？”对曰：“臣之所造能倡者。”穆王惊视之，趣步俯仰，信人也。巧夫，頷其颐，则歌合律；捧其手，则舞应节。千变万化，惟意所适。王以为实人也。

English:「*King Mu of Chou made a tour of inspection in the west...and on his return journey, before reaching China, a certain artificer, Yen Shih by name, was presented to him. The king received him and asked him what he could do. He replied that he would do anything which the king commanded, but that he had a piece of work already finished which he would like to show him. 'Bring it with you tomorrow', said the king, 'and we will look at it together.' So next day Yen Shih appeared again and was admitted into the presence. 'Who is that man accompanying you?' asked the king. 'That, Sir', replied Yen Shih, 'is my own handiwork. He can sing and he can act.' The king started at the figure in astonishment. It walked with rapid strides, moving its head up and down, so that anyone would have taken it for a live human being. The artificer touched its chin, and it began singing, perfectly in tune. He touched its hand, and it began posturing, keeping perfect time. It went through any number of movements that fancy might happen to dictate. The king, looking on with his favourite concubine and other beauties, could hardly persuade himself that it was not real.[75]*」

— 列子・汤问

(Lieh-tzu・The Questions of T'ANG)

The epigraph tells a story that King Mu of Chou made a tour of inspection in the west, on his return journey, a man named Yen Shih presented a handiwork which could sing, act and made the King think it was a real man in astonishment[27, 75], which was recorded in one of the most famous Taoist teachings named Lieh-tzu finished two thousand years ago. Fig.1.1 shows the content of this story re-edited in Siku Quanshu two hundred years ago. During thousands of years, from east to the west, making an automation which can mimic human activities has attracted great interest of talents. And resulting to one famous note by Marvin Minsky about the intelligent and emotions of robots: the question is not whether intelligent machines can have any emotions, but whether machines can be intelligent without any emotions[69]

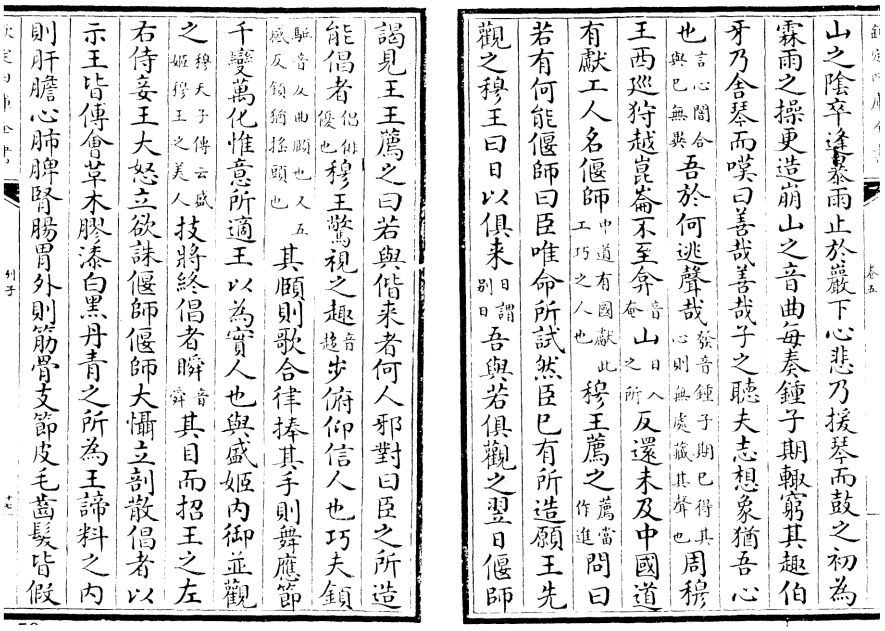

Figure 1.1: The story of King Mu of Chou and Yen Shih written in the Question of T'ANG recorded in Siku Quanshu

With the development of science, the two aspects of a robot: body and soul are improved in different ways. For the body, materials change from wood to metal or plastic, power improves from potential energy to electric source or gasoline engine, with powerful control algorithms, the body can be made as human beings more than ever. However, the 'soul' consisted of intelligent improve slowly and yet recent years get great advance due to the AI(Artificial Intelligence) innovation. In this thesis, we focus on enhancing the emotion recognition ability of 'soul', and giving an application demo of automatic

emotional expression based on an advanced humanoid robot REN-XIN.

In this thesis, the emotion recognition ability is based on text, and it is a text emotion classification problem, one of the key dimensions in "Affective Computing" provided by Picard in 1997, which is of great importance and is computing that relates to, arises from, or deliberately influences emotion or other affective phenomena[85]. The recognition model in our research is trained from a multi-label Chinese emotion corpus annotated by Ren Lab(Ren_CECps). In order to find the complex distributions of every annotated sentences, we make a 2D graph using t-SNE(t-Distributed Stochastic Neighbor Embedding) algorithm, and a 3D visualization map is also proposed. To avoid the point overlap, we propose an emotion separated TF $\cdot$ IDF(SeTF $\cdot$ IDF) algorithm to assign one multi-label annotated sentence with different feature vectors for every single category.

The 2D and 3D reduced distribution maps gives us a clearer view of the distance within every separated emotion sentence. The points position changed between SeTF $\cdot$ IDF and TF $\cdot$ IDF inspired us to apply a distance measurement algorithm to recognize the emotion categories. We finally propose a fast WMD method which is a 16000 times faster version of Word Mover's Distance(WMD) algorithm. Utilizing the distance features generated by fast WMD method, our experiments show that the SVM classifier get the best F1 scores of 0.318 than the features calculated by SeTF $\cdot$ IDF and TF $\cdot$ IDF of F1 scores of 0.293 and 0.203 respectively. Our cross-language experiments based on Chinese emotional corpus Ren_CECps and English news dataset 20 newsgroup show that with the fast WMD computed features, SVM classifiers get 3 times and 9 times improvement of F1 scores respectively compared with the same dimension features reduced from tradition TF $\cdot$ IDF.

Despite the huge progress achieved in robot field, the expression controls of humanoid robot with visual human-likeness face are still manually operated by developer for specified or limited scenarios. With the 'soul' embedded with the emotion recognition model trained with the distance features above, we try to enhance the 'body' of our humanoid robot REN-XIN by improve the expression ability. We utilized the proposed fast WMD[94] method which can recognize nine emotion categories in texts as emotional trigger to generate the corresponding action labels according to the robot's response. For the robot system, running the computed basic expression and the voice at the same time, we can get an

acceptable humanoid robot interaction with emotion expression.

During the running interaction with Actroid REN-XIN, the fast WMD based emotional trigger system needs at least 7s to deal with the response. To make a real time interaction, the seamless user experience is an essential aspect. Thereby, for people are communicating with humanoid robot, the delayed feedback results to no long communication desire. To solve this rough gap, we propose a CNN+LSTM based DNN model. In the experiments, we utilize the same sub-data sets of the Chinese emotional corpus(Ren_CECps) used in fast WMD experiments which are split in two ways: one is 50% for training and 50% for testing(1v1 experiment); the other one is 80% for training and 20% for testing(4v1 experiment). The experiments are proceeded in fast WMD, CNN+LSTM, CNN and LSTM respectively. The results show that CNN+LSTM gets the best result of F1 score 0.35 in 1v1 experiment, and almost the same accuracy with fast WMD of F1 scores 0.367 with 0.366 in 4v1 experiment. In the training process, our experiments show that the DNNs only need 3 epochs to finish training. This is not only the difference between minutes and weeks cost in training, but also the extended flexibility for the actroid robot. The CNN+LSTM model is not all good, it still has weaknesses. Those will be discussed in the final of this thesis.

1.1 Thesis Organization

This thesis covers the theories, methods, results, and discussions about the complex emotion prediction and emotion-related topic development, which is organized in the rest chapters as follows.

Chapter 2: Gives some basic knowledge about machine learning, feature selection and deep neural network. In this chapter, the SVM, CNN and RNN will be specially introduced. The emotion corpus used in this thesis will also be described.

Chapter 3: Presents the social background of Affective Computing, exporting the key research field of emotion computing and sentiment analysis. Relying on the technology of robotic, the development of humanoid robot is also introduced.

Chapter 4: Describes the Visualization of Ren_CECps. This chapter is the key part of the thesis. The algorithm and deep learning structures learned are all inspired by the

visualization results.

Chapter 5: Propose two emotion classification methods: one is Distance Features based machine learning model trained with SVM, gets the best results compared with traditional $TF \cdot IDF$ and emotion separated $TF \cdot IDF$. The other one is deep neural network based emotion classification model, multi-label and multi-class learning will be both explored in this part.

Chapter 6: Describes some related tasks which mainly takes temporal informations into account. We are trying to verify whether temporal informations can improve emotion recognition for time sensitive questions.

Chapter 7: Gives the application of our Emotional Triggers System for Humanoid Robot REN-XIN. This chapter describes the experiments that utilizing our proposed emotion classification models to enhance the interaction with robot. To make a more smooth interaction, the CNN_LSTM based model will be discussion in this part.

Chapter 8: Shows the evaluation process of emotion classification and emotional trigger systems.

Chapter 9: Presents the discussion of the experiments and gives the conclusion and future works.

Chapter 2

Background

In this chapter, some basic knowledge of Natural Language Processing, Deep Learning and traditional machine learning models will be introduced.

2.1 Natural Language Processing

Natural language processing is a reach field that explores how computers can be used to understand and manipulate natural language text or speech to do useful things[17]. Unlike the words in English or other Romance languages are naturally split with space, the words in Chinese are all written continuously without any natural segment tags, except the punctuations. This means the word based algorithms applied in English easily must be faced with the fact that there are no words in Chinese, only characters can be afforded. but characters only model will cause semantic loss and on the contrary get a worse performance on words based models.

Word Segmentation Or called participle, is the problem of dividing a string of written language into its component words. For example, converting a string $c_1c_2c_3c_4c_5c_6$ to space split format $c_1 \quad c_2c_3 \quad c_4c_5c_6$, where c_i means characters and $\{c_i\}$ means the words. Before the word segmentation algorithms become maturity, the most used algorithm for preparing the segmentation text is n-gram model, which can turn one string into several n-width sub-strings. With the development of statistical machine learning, maximum matching word segmentation[115] and smi-markov conditional random field(smi-CRF) for sequence

segmentation[3] are proposed respectively. In this thesis, we use a HHMM based Chinese segmentation tool named ICTCLAS[1] or PyNLPIR(a python version of ICTCLAS)[2] for the preprocessing of text[121].

TF · IDF As a short of term frequency-inverse document frequency[41], is a numerical statistic algorithm in information retrieval which is intended to show how important a word is to a document in one corpus[54]. The TF · IDF is normally computed as follows:

$$\begin{aligned} tf_{ij} &= \frac{tf_{ij}}{\sum_j tf_{ij}} \\ idf_{ij} &= \log \frac{N}{df_i + 1} \\ tfidf_{ij} &= tf_{ij} \times idf_{ij} \\ where, \quad & i, j \in \mathbb{N} \end{aligned} \tag{2.1}$$

where, tf_{ij} means the frequency of $word_j$ in the i^{th} document. $\sum_j tf_{ij}$ means the total frequencies of words in the i^{th} document. N means the total number of documents in corpus, df_{ij} means the count of how many documents contain the $word_j$ and the $df_i + 1$ is to avoid division by zero. In practice, we use document-term(D-T) matrix to calculate the TF · IDF as shown in Table 2.1.

Table 2.1: The example of document-term matrix

	$word_1$	$\cdots$	$word_j$	$\cdots$	$word_m$
D_1	tf_{11}	$\cdots$	tf_{1j}	$\cdots$	tf_{1m}
:	:	$\ddots$	:	$\ddots$	:
D_i	tf_{i1}	$\cdots$	tf_{ij}	$\cdots$	tf_{im}
:	:	$\ddots$	:	$\ddots$	:
D_n	tf_{n1}	$\cdots$	tf_{nj}	$\cdots$	tf_{nm}

In which, D_i means the i^{th} document in corpus and $word_j$ means the j^{th} word in vocabulary. tf_{ij} means the frequency of $word_j$ in document i. For idf computing, the D-T matrix only need to convert tf_{ij} into 0 if $tf_{ij} = 0$, or 1 for otherwise. Then summing the column j to get the df of $word_j$. In this thesis, all of the TF · IDF are calculated by

[1] http://ictclas.nlpir.org/
[2] https://github.com/tsroten/pynlpir

the "text"[3] package of scikit-learn[81].

Word2vec The word2vec[4] is a tool published by Mikolov et al. which contains model and application that can convert words into n-dimension vectors by training a hidden layer network. It has two traditional models to train the word vectors: one is the continuous bag-of-word model(CBOW)[65], the other one is the skip-gram model[65, 67].

We assume the target word is K_c, its i nearby words are $K_{c-i}, \cdots, K_{c-1}, K_{c+1}, \cdots, K_{c+i}$, and all of the words are a $V-dimension$ one-hot vector defined as ν. The hidden layer is represented as H. Its a $N-dimension$ vector and h_i means the i^{th} unit. Applying those, the training process can be simply defined as two linear transformation as follows, though the output layer has a nonlinear softmax model.

$$\begin{aligned} H &= W^T_{VxN} \cdot \nu \quad , for \quad Input \quad layer \rightarrow Hidden \quad layer \\ \nu &= W^{out}_{NxV} \cdot H \quad , for \quad Hidden \quad layer \rightarrow Output \quad layer \end{aligned} \tag{2.2}$$

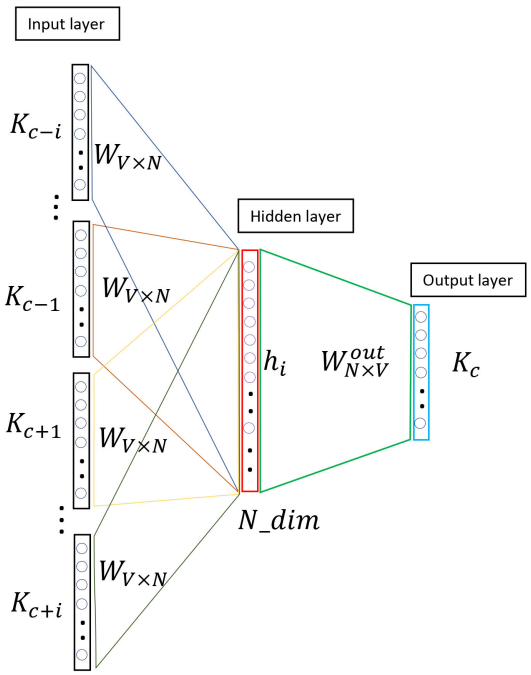

Figure 2.1: The continuous bag-of-word model

[3]http://scikit-learn.org/stable/modules/classes.html#module-sklearn.feature_extraction.text
[4]https://code.google.com/archive/p/word2vec/

Fig.2.1 and Fig.2.2 show the two models in simple networks with corresponding weights matrices. In the figures, we can clearly find that the CBOW model uses the i neighbors as input and the target word as label sample. While the skip-gram model goes the opposite way, utilizing the target word as input and the i neighbors as label samples.

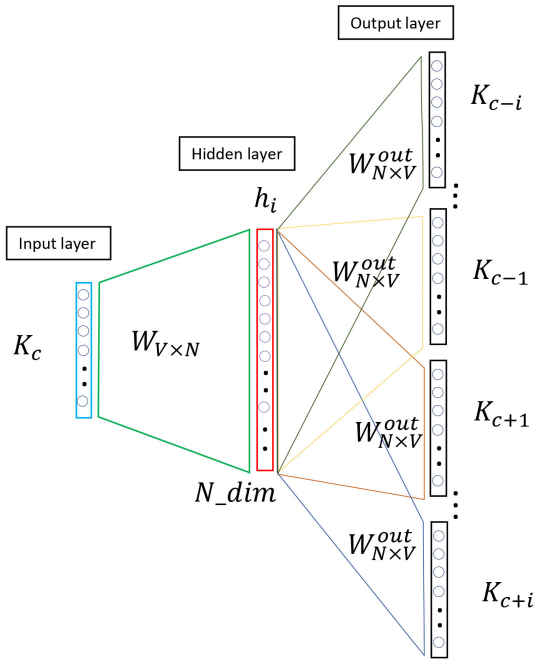

Figure 2.2: The skip-gram model

Ren_CECps The Chinese emotional corpus annotated by Ren Lab is also called Ren_CECps[5][87]. It contains 1487 blogs crawled through Internet. Every blogs are annotated by the document→ paragraph→sentence structure with eight emotion tags of ”Joy”, ”Hate”, ”Love”, ”Sorrow”, ”Anxiety”, ”Surprise”, ”Anger”, ”Expect”.

Fig.2.3 is the sample of document→ paragraph→sentence structure annotation for the blogs in Ren_CECps. As shown in Fig.2.3, the three level annotation structure can be clearly tagged by the $< document >$ tag, $< paragraph >$ tag and $< sentence >$ tag respectively. Each levels is annotated with eight emotion categories and its corresponding intensities, the intensity is a discrete value between 0.0 and 1.0. For the title of document, the polarity is annotated except the emotion categories. In each of $< document >$ and $< paragraph >$ levels, the topics are also targeted in the value of $< Topic >$ tag following

[5]http://a1-www.is.tokushima-u.ac.jp/member/ren/Ren-CECps1.0/DocumentforRen-CECps1.0.html

with the eight emotion tags.

```
<document>
        <Joy><Hate><Love><Sorrow>
        <Anxiety><Surprise><Anger><Expect>
        <Topic>
        <title>
                <Keywords>
                <E_phrase>
                <punctuation>
                <Polarity>
                <Joy><Hate><Love><Sorrow>
                <Anxiety><Surprise><Anger><Expect>
        </title>
        <paragraph>
                <Joy><Hate><Love><Sorrow>
                <Anxiety><Surprise><Anger><Expect>
                <Topic>
                <sentence>
                        <Keywords>
                        <degree_adv>
                        <E_conjunction>
                        <E_phrase>
                        <Rhetoric>
                        <Opinion_Fact>
                        <punctuation>
                        <Polarity>
                        <Opinion_holder>
                        <Opinion_target>
                        <Joy><Hate><Love><Sorrow>
                        <Anxiety><Surprise><Anger><Expect>
                </sentence>
                ...
                ...
        </paragraph>
        ...
        ...
</document>
```

Figure 2.3: The sample of three level annotation structure of Ren_CECps

The sentence level is the most important level to annotate for almost the whole emotions are contained in this level. The polarity and eight emotion categories of every sentences are annotated as default, the emotional keywords and phrases are tagged in the tags of $< Keyword >$ and $< E\_phrase >$ the same as the tags in $< title >$. Specially, the eight emotion categories and corresponding intensities are annotated for word-based emotion research which are not showed in Fig.2.3. Some linguistics informations are annotated to extend the use of this corpus, like degree adverb tagged as $< degree >$, punctuation as $< punctuation >$ and rhetoric as $< Rhetoric >$. For opinion mining, the opinion holder and opinion target are annotated in tags of $< Opinion\_holder >$ and $< Opinion\_target >$ respectively. In $< punctuation >$ tag of title and sentence, the emotion type is added if possible.

Table 2.2 shows the sentences numbers with different labels in Ren_CECps. The percentages of sentences with more than three emotions is less than 0.5%, sentences with one or two emotions take the percentage of over 94%, sentences with one or two or three emotions get the percentage of over 99%. So, in general, a sentence has no more than 3 emotions.

Table 2.2: The number of multi-label sentences in Ren_CECps

label No.	total	one	two	three	four	five	six
sentence No.	36525	22751	11731	1847	175	15	6
per. (%)	100	62.2888	32.1177	5.0568	0.4791	0.0004	0.0001

2.2 Deep Neural Networks

Deep neural networks are almost the most popular feature learning structure in every machine learning fields. The key layers of DNN are all based on recurrent neural network(RNN) and convolutional neural network(CNN) respectively or both in the traditional structure or improved ones.

2.2.1 CNN

Convolutional neural network was published famously for its backpropagation algorithm[52] and gradient-applied learning[53] in handwritten zip code recognition. But soon met its winter in the 1990's for lack of data. After Imagenet was published in 2009[21], the deep neural networks[49] came again back to the top and open a new AI era.

In this sub-part, The traditional CNN will be introduced by a simple structure which was shown in Fig.2.4. Although CNN is most used in Image field based on pixel, it can also be applied to neural language processing based on the feature vectors of text. The Fig.2.4 is drawn following the pixel style, where every units in the input layer represent the pixel of image. The Fig.2.4 shows a 28×28 input image with one convolution kernel of size 5×5, the stride in this example is 1,so the next layer gets a 24×24 scaled feature map. The next is a pooling layer with size of 6×6, calculated by a 4×4 pooling operation. The final is output layer with full connection from pooling layer, the output is a 8D vector which can be a multi-class classifier or a multi-label classifier depending on the task.

Assuming the input image X^1 is a $n \times n$ matrix with one channel and $x^1_{i,j}$ means the elements. The convolution kernel K is a $k \times k$ matrix and $k_{u,v}$ represents the values, where $k < n$. The feature map X^2 calculated by K is a $m \times m$ matrix, and $x^2_{i,j}$ means elements. The pooling map P is a $t \times t$ matrix with elements of $x^3_{i,j}$, calculated by a $r \times r$ size pooling operation. The final layer Y is a $w - dimension$ vector, y_i is the value. For convolution and pooling layers, $f(x_{i,j})$ means the output matrix and the convolution layer will not include activation functions.

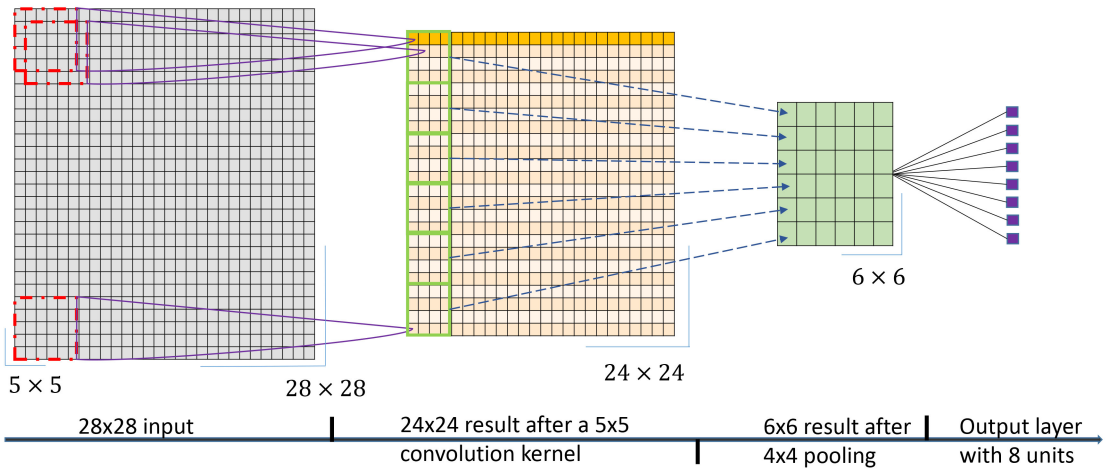

Figure 2.4: The construction of CNN with parameters

Applying the representation above, the CNN in Fig2.4 can be calculated as follows:

The forward propagation:

- Convolution

$$x^2_{i,j} = \sum_{u}^{k} \sum_{v}^{k} x^1_{i+u,j+v} \cdot k_{u,v}$$
$$X^2 = f(x^2_{i,j}) \tag{2.3}$$

- Pooling

$$x^3_{i,j} = \sum_{ir}^{(i+1)r-1} \sum_{jr}^{(j+1)r-1} x^2_{i,j}$$
$$P = f(x^3_{i,j}) \tag{2.4}$$

- Output

$$
\begin{aligned}
Y &= W \cdot (Flat(P))^T + b \\
C &= softmax(Y) \\
where, C \quad means \quad predicted \quad labels.
\end{aligned}
\tag{2.5}
$$

The backward propagation:

Let δ be the error for the final layer in CNN with a cost function $J(W, b)$, the error for the pooling layer is computed as

$$
\delta^2 = upsample((W^2)^T\delta) \cdot f^{'}(x^2_{i,j}) \tag{2.6}
$$

the *upsample* operation has to propagate the error through the pooling layer by calculating the error to each unit incoming to the pooling layer. To calculate the gradient to the convolution kernel, the formulas are as

$$
\begin{aligned}
\nabla_{W^1} J(W, b) &= \sum_{i,j} (x^1_{i,j}) * rot90(\delta, 2) \\
\nabla_{b^1} J(W, b) &= \sum \delta
\end{aligned}
\tag{2.7}
$$

For the backward propagation, the next subsection will give a specific mathematic explanation based on RNN structure.

2.2.2 RNN

Recurrent neural network(RNN) was published for a long time[98, 86]. The experiments in language model based RNN show the excellent ability of RNN to handle time sequence problems[66]. Fig.2.5 shows a traditional RNN unit and its unrolled sequence structure in time.

In Fig.2.5, X, Y and H means the input, output and hidden status respectively. X^t, Y^t and H^t means the corresponding t-time values. We assume x^t_i, y^t_i and h^t_i to represent the cell of the three layers. W_{in} means the weight matrix of input to hidden layer, W_h means the self transfer matrix of hidden layer and W_{out} means the weight matrix for hidden layer

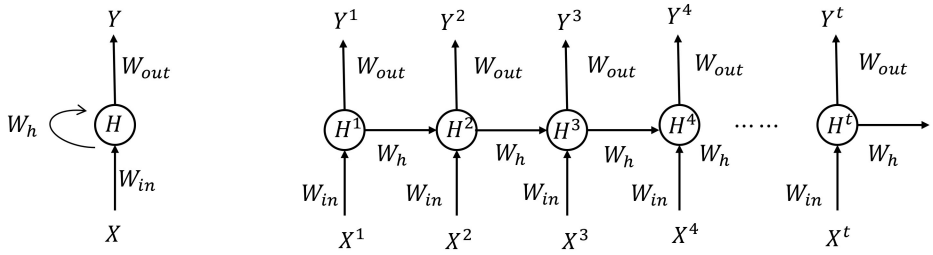

Figure 2.5: The traditional RNN structure(left) and its unrolled sequence in time(right)

to output layer. U_{in} and U_{out} represents the input and output of hidden layer with the cell at t-time represented as $(u_{in}^i)^t$ and $(u_{out}^i)^t$. Relaying on these, the forward propagation of traditional RNN can be calculated as follows:

Forward propagation:

$$\begin{aligned} U_{in} &= W_{in}X \\ H^t &= \tanh(W_h H^{t-1} + U_{in}) \\ U_{out} &= W_{out}H^t \\ Y &= U_{out} \end{aligned} \tag{2.8}$$

where, $Y = U_{out}$ is satisfied for only no activation function added, if the output layer has an activation function(like softmax), the equation will be $Y = \sigma(U_{out})$, in which σ means the activation function. In this thesis, the RNN will be explained in a simple way without activation.

Loss function: Assuming that T represents the truth label of samples, the loss function in neural networks have two common styles:

- Sum of squared error(quadratic error)

$$E = \frac{1}{2}(T - Y)^2 \tag{2.9}$$

- Cross entropy error

$$E(T, Y) = -[T\ln(Y) + (1 - T)\ln(1 - Y)] \tag{2.10}$$

for different output by formula 2.9 and formula 2.10, the derivatives of the output layer are as follows:

- derivation of Quadratic error

$$\frac{\partial E}{\partial Y} = \frac{\partial}{\partial Y}\frac{1}{2}(T-Y)^2 = Y - T \tag{2.11}$$

- derivation of Cross entropy error

$$\frac{\partial E}{\partial Y} = \frac{\partial}{\partial Y}(-[T\ln(Y) + (1-T)\ln(1-Y)]) = \frac{Y-T}{Y(1-Y)} \tag{2.12}$$

In the backward propagation, for a simple explanation, we only take quadratic error in consideration.

Backward propagation:

- The RNN has no activation for output layer, so the gradients for the output of output layer and the input of the output layer are the same in formula 2.11. As in formula 2.8, $Y = U_{out}$.

- Next we need to calculate the gradient of hidden layer to output layer, as in formula 2.8, $U_{out} = W_{out}H^t$.

$$\begin{aligned}\frac{\partial E}{\partial W_{out}} &= \frac{\partial E}{\partial U_{out}} * \frac{\partial U_{out}}{\partial W_{out}} \\ &= \frac{\partial E}{\partial Y} * \frac{\partial W_{out}H^t}{\partial W_{out}} \\ &= (Y-T) * H^t\end{aligned} \tag{2.13}$$

- The hidden layer H^t is not only connected with the U_{out}, but also contributed for the next hidden status H^{t+1} by self updating formula $U_{hidden}^{t+1} = W_h H^t + W_{in}X$. So the derivation of H^t based on the loss is the sum of these two derivations, as follows:

$$\begin{aligned}\frac{\partial E}{\partial H^t} &= \frac{\partial E}{\partial U_{out}} * \frac{\partial U_{out}}{\partial H^t} \\ &= \frac{\partial E}{\partial Y} * \frac{\partial W_{out}H^t}{\partial H^t} \\ &= (Y-T) * W_{out}\end{aligned} \tag{2.14}$$

$$\begin{aligned}\frac{\partial E}{\partial H^{t+1}} &= \frac{\partial E}{\partial U_{hidden}^{t+1}} * \frac{\partial U_{hidden}^{t+1}}{\partial H^t} \\ &= \frac{\partial E}{\partial U_{hidden}^{t+1}} * \frac{\partial (W_h H^t + W_{in} X)}{\partial H^t} \\ &= \frac{\partial E}{\partial U_{hidden}^{t+1}} * W_h\end{aligned} \tag{2.15}$$

the final updated derivation of H^t is in formula 2.16:

$$\begin{aligned}\frac{\partial E}{\partial H^t} &= \frac{\partial E}{\partial H^t} + \frac{\partial E}{\partial H^{t+1}} \\ &= (Y - T) * W_{out} + \frac{\partial E}{\partial U_{hidden}^{t+1}} * W_h\end{aligned} \tag{2.16}$$

- The derivation of input for self updating of hidden layer can be calculated as follows:

$$\begin{aligned}\frac{\partial E}{\partial U_{hidden}^t} &= \frac{\partial E}{\partial H^t} * \frac{\partial H^t}{\partial U_{hidden}^t} \\ &= \frac{\partial E}{\partial H^t} * \frac{\partial \tanh(U_{hidden}^t)}{\partial U_{hidden}^t} \\ &= \frac{\partial E}{\partial H^t} * (1 - (\tanh(U_{hidden}^t))^2) \\ &= \frac{\partial E}{\partial H^t} * (1 - H^t \odot H^t) \\ where, \quad H^t &= \tanh(U_{hidden}^t)\end{aligned} \tag{2.17}$$

- The derivation of the weight for input layer to hidden layer can be calculated as follows:

$$\begin{aligned}\frac{\partial E}{\partial W_{in}} &= \frac{\partial E}{\partial U_{in}} * \frac{\partial U_{in}}{\partial W_{in}} \\ &= \frac{\partial E}{\partial U_{in}} * X\end{aligned} \tag{2.18}$$

- The derivation of the weight for hidden layer to hidden layer in time sequence can be calculated as follows:

$$
\begin{aligned}
&Given \quad U^t_{hidden} = W_h H^{t-1} + W_{in} X \\
&\frac{\partial E}{\partial W_h} = \frac{\partial E}{\partial U^t_{hidden}} * \frac{\partial U^t_{hidden}}{\partial W_h} \\
&\qquad = \frac{\partial E}{\partial U^t_{hidden}} * H^{t-1}
\end{aligned} \tag{2.19}
$$

According to all these formulas, we can update the weight matrices by gradient descent below:

$$
\begin{aligned}
&\frac{\partial E}{\partial W_{out}} = (Y - T) * H^t \\
&\frac{\partial E}{\partial W_{in}} = \frac{\partial E}{\partial U_{in}} * X \\
&\frac{\partial E}{\partial W_h} = \frac{\partial E}{\partial U^t_{hidden}} * H^{t-1} \\
&\frac{\partial E}{\partial H^t} = (Y - T) * W_{out} + \frac{\partial E}{\partial U^{t+1}_{hidden}} * W_h \\
&\frac{\partial E}{\partial U^t_{hidden}} = \frac{\partial E}{\partial H^t} * (1 - H^t \odot H^t)
\end{aligned} \tag{2.20}
$$

Fig. 2.6 shows the gradient flow of RNN, where red dot line means the direction of gradient in the RNN unit.

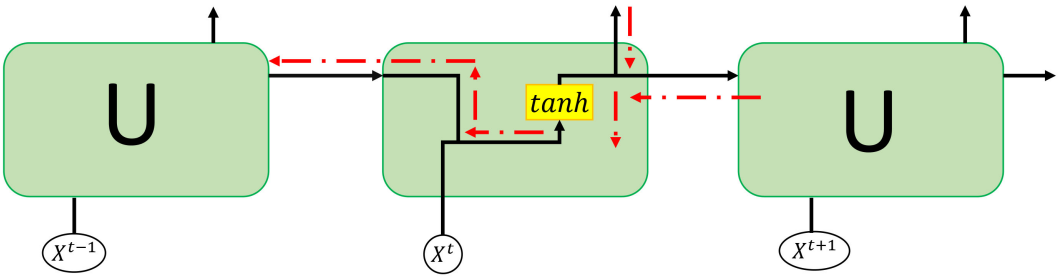

Figure 2.6: The gradient flow of RNN

The gradient flowing in the RNN unit will face the gradient vanish problem while the iteration times increase. That means after several times, the RNN unit can not catch any information. In order to enhance the long term memory, Long Short-Term Memory(LSTM) was proposed[36]. Fig. 2.7 shows the structure of LSTM, the gradient flow by red dot lines is also annotated in this figure. One of the other famous new RNN structures is Gated Recurrent Unit(GRU)[15], which was first proposed for machine translation. The GRU will not be introduced in this thesis.

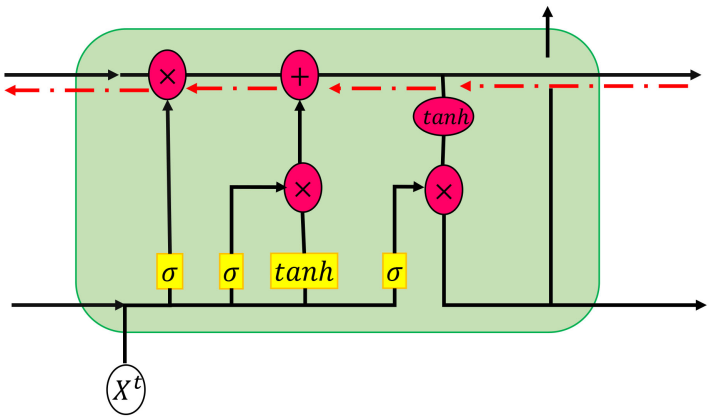

Figure 2.7: The gradient flow of LSTM

2.3 Traditional Classifiers in Machine Learning

Most of the traditional machine learning models are statistics based algorithm, like expectation maximization (well known for EM algorithm)[72], naive Bayes classifier[95, 120]. But support vector machine(SVM) classifier is not, while it has excellent ability for non-linear classification with light structure and little memory relay[18].

SVM algorithm utilizes the kernel function(for example, polynomial kernel) to match the low dimension data into high dimension space, in which the linear inseparable data can be line separable. As show in Fig.2.8.

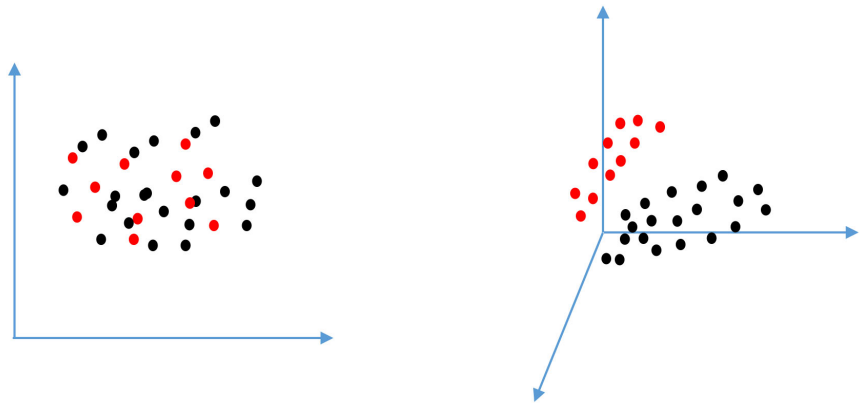

Figure 2.8: The example of linear inseparable points in 2D(left) and their linear separable positions in 3D(right)

Fig.2.8 is a example of matching 2D data into 3D space, and the original low dimensional data in high dimension space can clearly be separated by a hyperplane. Assuming the kernel for matching is $\phi(\cdot,\cdot) : R^{low} \rightarrow R^{high}$. Defined the samples in low space as (x_i, y_i), the hyperplane in high dimensional space can be defined as follows:

$$f(x) = \sum_{i=1}^{n} a_i y_i < \phi(x_i), \phi(x) > +b \tag{2.21}$$

where, $< \phi(x_i), \phi(x) >$ means the inner product of two points in high matched dimension space.

In kernel function, the innner product of matched high dimensional points equals to the kernel result of the inner product of the two point in original low dimensional space, as formula 2.22:

$$< \phi(x_i), \phi(x) >= \phi < x_i, x > \tag{2.22}$$

Thus, the hyperplane can be written as:

$$f(x) = \sum_{i=1}^{n} a_i y_i \phi < x_i, x > +b \tag{2.23}$$

while the parameter a_i can be approximated as dual problem:

$$\begin{aligned} \max_{a} \quad & \sum_{i=1}^{n} a_i - \frac{1}{2} \sum_{i,j=1}^{n} a_i a_j y_i y_j \phi < x_i, x_j > +b \\ s.t., \quad & a_i \geq 0, \quad i = 1, \cdots, n \\ & \sum_{i=1}^{n} a_i y_i = 0 \end{aligned} \tag{2.24}$$

The kernel function has two common types:

- Polynomial kernel: $\phi < x_i, x_j >= (< x_i, x_j > +R)^d$
- Gaussian kernel: $\phi < x_i, x_j >= \exp(-\frac{\|x_i - x_j\|^2}{2\sigma^2})$

Chapter 3

Related Work

Since the pace of modern life becomes more and more fast, people always work and live with high stress. From the report published by WHO, one in four people in the world will be affected by mental or neurological disorders at some point in their lives[80]. Thus, it's momentous to make emotion computable for psychotherapy, health prediction or any other fields.

Emotions play an important role in successful and effective human-human communication[11]. There is also significant evidence that rational learning in humans depends on emotions[85]. With Google AI computer program AlphaGo beat Jie Ke at a three-game match in the 2017 Future of Go Summit, artificial intelligence drew the attention of the globe once again and will continue standing on top of the tides. This makes us recall the famous words noted by Marvin Minsky about the future of emotion computing: the question is not whether intelligent machines can have any emotions, but whether machines can be intelligent without any emotions[69].

Accompanying with the blossoming of the Word Wide Web, it's much easier to obtain text data to train a classifier. To show the abundant features of data, some interactive visualization methods were presented, like the most used parallel coordinates[39] and scatter-plot matrix[25] in attribute-decided data visualization. For the uncertainty of data labels, the measurement can be got in term of probabilities[102], which is useful in unTangle Map[13] for multi-label data visualization. As machine learning algorithms were introduced into NLP, a lot of annotated corpus without specific attribute values can be visualized by dimension scaling[9, 74],SVD[103],t-SNE[62]. With better visualiza-

tion, the classification models can also be enhanced by integrating visual features and text features[61, 124, 35]. When it comes in large graph visualization, avoiding notes overlapping is another hot research topic. The principal method to solve this situation is elongating the distance within points, like force transfer[38] or changing the distribution of categories[4]. This is exactly what we do in this paper.

For similarity computing using a metric between two distributions, the Earth Mover's Distance(EMD)[96] is one of the well-studied algorithms. By calculating the minimum cost that transform the distributions of color and texture into the other, the EMD can get better results for content-based image retrieval[97] and even can detect phishing web pages by visual similarity[26].

The most commonly used algorithms to represent the documents for similarity computing are statistic based algorithms like $\mathrm{TF} \cdot \mathrm{IDF}$[100], LDA[8], or trained vectors using deep neural networks[67, 51].In paper[107], Wan applied EMD into document similarity measurement successfully by decomposing the documents into a set of subtopics and using EMD to evaluate the similarity of many-to-many matching between the subtopics.

Limited by the NLP and machine learning algorithms, the pioneering studies in emotion computing were based on lexicons[2, 5, 12]. After years' development, several annotated multi-emotion corpus were published [68, 87, 73]. Based on the emotion annotated corpus, the derived lexicon with multi-emotion tags can get higher F1 scores compared with traditional lexicon-based feature[55]. Relying on those emotional corpus, sentiment analysis can have a sub-field of emotion computing. A lot of machine learning algorithms were explored. SVM, Naive Bayes and Maximum Entropy are some of the most common algorithms used[88, 110, 108]. Some research using HMMs had also achieved better results[89, 84].

Emotion computing in Chinese has attracted many researchers due to the development of microblogging and tweet. Some studies in sentiment analysis of Chinese documents[105] turn to emoticon-based sentiment analysis[125]. But the studies of hidden sentiment association in contents[104, 60] are still one of the key points, like Chinese idiom emotion recognition[111], and can be especially important for measuring the mental healthy of humanity[93, 56]. To improve the study of affective computing in social networks, some standard corpus based on weibo data had been published[33, 40, 14]. Inspired by the ex-

cellent performance of deep neural network in image recognition, a lot of researches based on RNN[123], LSTM[119], CNN[117] for sentiment analysis had been done, works based on sentiment embeddings also get excellent results[106] and will attract more and more attention.

For estimating emotion of words that not registered in the lexicon, EMD can be applied to vectors of words and get higher actuary compared with using only word importance value[64].As the high time-consuming in EMD, the EMD based methods always limited into keywords or topics, not the full words. With the fast specialized solvers of EMD[82] was published, words fully transformed experiments can be carried out[50].

The early methods used in emotion computing are the same with machine learning field, like LDA[92], SVM[118], Naive Bayes[101],is one of the most active research areas in NLP field[57]. Most of the works are focused on social networks like twitter[71, 70], blogs[87, 59], microblog[113], or aimed to e-commerce purpose like movie reviews[79], products opinions[29].Limited to the scale of annotated corpus, the previous studies always go from the emotional lexicons based methods[12, 37]. The annotated emotion categories also vary according to the different corpus: from two(positive, negative) to four emotions(anger, fear, sadness and joy)[19] or eight emotions(Anger, anxiety, expect, hate, joy, love ,sorrow and surprise)[87]. These non-uniform annotated methods make this research field harder to extend large-scale corpus.

Recent years, deep neural network achieve high progress, and attract a lot of attentions. Deep learning algorithms can enhance sentiment classification on social data[112], and can handle large-scale sentiment classification[28]. Complex deep neural network applied on emotion cause extraction can also achieve great results[32]. The word2vec is one of the most famous networks for word embedding[65]. A suite of deep belief network had been proved effective for emotion recognition owing to its learned high-order nonlinear relationships[48]. Some works focused on character-level learning, and trained a character-level convolution networks[122, 47].A sentiment treebank proposed for semantic Compositionality[113] moves the field running into deep learning era.

A lot of deep neural networks based method have been proposed year by year. A CNN based neural network can achieve stat of the art results within little hyperparamter tuning[46], a neural network trained for a dialogue generation joining with emotion encoder[77],

a simple LSTM model for affect recognition framework[114], and also a combined model CNN-RNN can apply for video emotion recognition[22]. For more refined word representation, Glove was proposed[83]. Also many new deep neural networks are presented: like GAN[30], ResNet[76] and sequence to sequence model for emotional chatting[127].

With time flowing, the innovation of automation turns into the robot developments, and due to the technology improvement in decades, many humanoid robots were published for societies, some of them are WABOT by Waseda University, Asimo by Honda, Robonaut by NASA, Nadine by Nanyang Technological University and Actroid by Kokoro. Some of the humanoid robots have the same body structure and they can walk, hold objects, run or jump[63]. Some of the humanoid robots have human-like faces, and these robots can sing, speak languages and make expression, one of them named Sophia developed by Hong Kong-based company Hanson Robotics even become a Saudi Arabian citizen.

Despite the huge progress achieved in robot field, the expression controls of humanoid robot with visual human-likeness face are still manually operated by developer for specified or limited scenarios. To extend the application scenes,one way for solving this is to annotate nine basic expression actions for our Actroid robot, and when comes to interaction, we utilized the proposed fast WMD[94] method which can recognize nine emotion categories in texts as emotional trigger to generate the corresponding action labels according to the robot's response. For the robot system, running the computed basic expression and the voice at the same time, we can get an acceptable humanoid robot interaction with emotion expression.

Since the robot were created, making them more engaged has become one of the topic research fields. Marian et al.[6] deployed a real time facial expression system in the Aibo robot and the RoboVie robot to enhance user enjoyment. Diego et al.developed a framework to recognize human emotions through facial expression for NAO[23].To make a real time facial expression on humanoid robot REN-XIN, a forward kinematics model was proposed[91]. Building a whole-length is expensive, some groups try to very their facial expressions system on head only robot.

K Berms and J Hirth[7] utilized 6 basic facial expressions for humanoid robot head ROMAN, this was a behavior based control system. Hashimoto et al. developed a face robot for rich facial expression, this face robot has 18 control points and can easily imitate

six typical facial expressions[34]. A quick application of facial expression with head-neck coordination is employed on robot SHFR-III[45]. Another way to expression emotions for a robot partner: iPhonoid-B is to combine the facial and gestural together, which is made up with smart phone and servos [116].

Chapter 4

Visualization of Ren_CECps

For a multi-class corpus, we can make a 2D or 3D scatter-plot to have a good view of the distribution of the data. But for multi-class data with multi-label, to do the visualization with the same different colored points will make the 2D or 3D graph unreadable. Thus, how to match the multi-label information into a 2D or 3D graph is the key target need to be covered.In this paper, for visualizing the multi-label emotional corpus Ren_CECps, we propose an emotion separated TF $\cdot$ IDF method(SeTF $\cdot$ IDF) to represent each emotional category independently with different values. And to make a better 2D visualization, we use one of the stat-of-art dimension reduction algorithm t-SNE[62] to measure the low dimension distribution of Ren_CECps. And the sentences without emotional labels are regarded as 'neutral' category.

4.0.1 t-SNE

t-SNE(t-Distributed Stochastic Neighbor Embedding) is a technique for dimensionality reduction that is particularly well suited for the visualization of high-dimensional datasets[62]provided by L.J.P van der Maaten and G.E Hinton. Compared with PCA algorithms, t-SNE computing the distributions of every nodes in the datasets and rebuilding the distribution of those nodes in two or three dimension space. To get the best approximate results, t-SNE uses KL divergence to measure the distance of the two distributions. In this paper, the t-SNE tool uses TSNE in sklearn[81] and the programm is

followed a guidance blog written by Alexander Fabisch[1].

4.0.2 Emotion Separated representation

As mentioned above, considering the 'neutral' label as one emotion category, the total number of emotional categories needed to be calculated is nine. The keyword $word_i$ of every sentences represented by TF $\cdot$ IDF can be calculated through formula 4.1.

$$tfidf = \frac{tf_i}{\sum tf_i} \times \log \frac{N}{df_i + 1}, i \in \mathbb{N} \tag{4.1}$$

In which, $tfidf$ means the TF $\cdot$ IDF result of $word_i$. tf_i means the term frequency of the calculated word. $\sum tf_i$ means the frequency of the total words. N means the total sentences number. df_i means the total number of sentences which contain $word_i$. From formula 4.1, the conclusion we can get is that no matter what the words in the sentence are, the feature vector calculated though formula 4.1 for every annotated emotion labels of one sentence will be the same without any distinguishing.

The good news is that in Ren_CECps the emotion keywords of a sentence are annotated. Thus, for an annotated emotion keyword, we calculate its $tfidf$ if and only if the emotion keyword has the given emotion category for a specific emotion label. In this way, we can generate a distinctive feature vector for each emotion label of a sentence. This method is named emotion separated TF $\cdot$ IDF(SeTF $\cdot$ IDF) method, and SeTF $\cdot$ IDF can be described as the formulas below:

$$tfidf_{e_j} = \frac{S_e tf_i}{\sum tf_i} \times \log \frac{N}{df_e + 1}, i \in \mathbb{N}, j \in [0, 8] \tag{4.2}$$

where,

$$e_j \in [joy, hate, love, sorrow, anxiety, surprise, anger, expect, neutral] \tag{4.3}$$

In which, $tfidf_{e_j}$ means the TF $\cdot$ IDF result of emotion keyword $word_i$ in emotion category e_j. $S_e tf_i$ means the term frequency of emotion keyword $wordi$ in emotion category e_j. df_e

[1] http://nbviewer.jupyter.org/urls/gist.githubusercontent.com/AlexanderFabisch/1a0c648de22eff4a2a3e/raw/59d5bc5ed8f8bfd9ff1f7faa749d1b095aa97d5a/t-SNE.ipynb

means the total number of sentences which contain emotion keyword $wordi$.

4.0.3 The Result of Visualization

Following the algorithm below, the words without emotional labels annotated are calculated through formula 4.1, for the words with emotional label annotated, those are calculated by formula 4.2. The steps for visualization are as follows:

Algorithm 1 The Procedure of Visualization

Require: S - sentences in Ren_CECps, L - labels of sentences
Ensure: 2D graph data

function TWO-DIMENSIONALIZATION(S, L)
  for $sentence \in S$ **do**
    calculate vectors by formula(1) and formula(2);
    add vector into matrix M and corresponding $label \in L$ into list l
  end for
  $M \xrightarrow{SVD} M_{50}$ ▷ M_{50} - 50 dimensions' row matrix reduced from M
  $M_{50} \xrightarrow{t-SNE} M_2$ ▷ M_2 - 2 dimensions' row matrix reduced from M_{50}
  return M_2, l
end function
function DRAW GRAPH(M_2, l)
  render set:
    $(Anger, Anxiety, Expect, Hate, Joy, Love, Neutral, Sorrow, Surprise)$
    $(red, blue, yellow, green, black, gray, orange, purple, pink)$
  return Graph
end function

The Fig.4.1 and Fig.4.2 show the visualization of Ren_CECps in TF·IDF and SeTF · IDF respectively. We can find that the overlapping points in TF·IDF have been separated in SeTF · IDF. There are also some conclusions we can get:

- The elongated distances between every category of sentences and the distributions changed in SeTF · IDF indeed have a better visual result compared with TF · IDF.
- "Love" points have the similar distribution compared with "Anxiety" points;
- Most of the "Sorrow" points come with no other emotional points embedding into their group.
- Sentences may have completely opposite emotion categories. In some clusters, there are pairs like "Sorrow and Joy", "Hate and Love".

- Fuzziness emotion categories such as "Expect" have the most frequency to appear with other emotion categories.

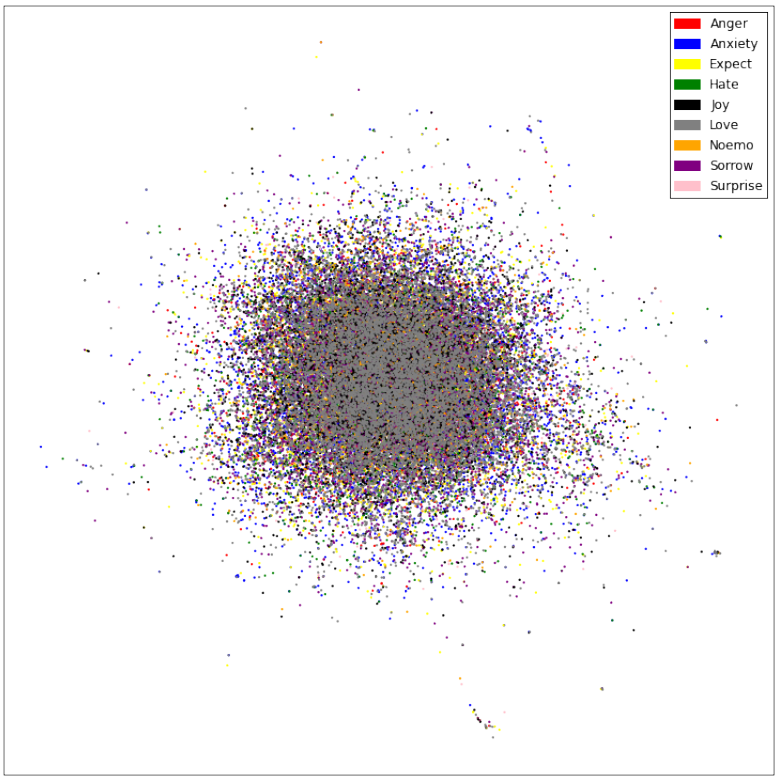

Figure 4.1: Visualization of Ren_CECps in traditional TF·IDF

Based on those features, we can have a more clear vision of Ren_CECps. Fig.4.3-Fig.4.10 show visualization of the traditional TF·IDF and the SeTF · IDF of Anger, Anxiety, Expect, Hate, Joy, Love, Sorrow and Surprise respectively. For a better view, we also make a 3D visualization result for the emotional corpus[2].

[2]http://a1-www.is.tokushima-u.ac.jp/data_all/

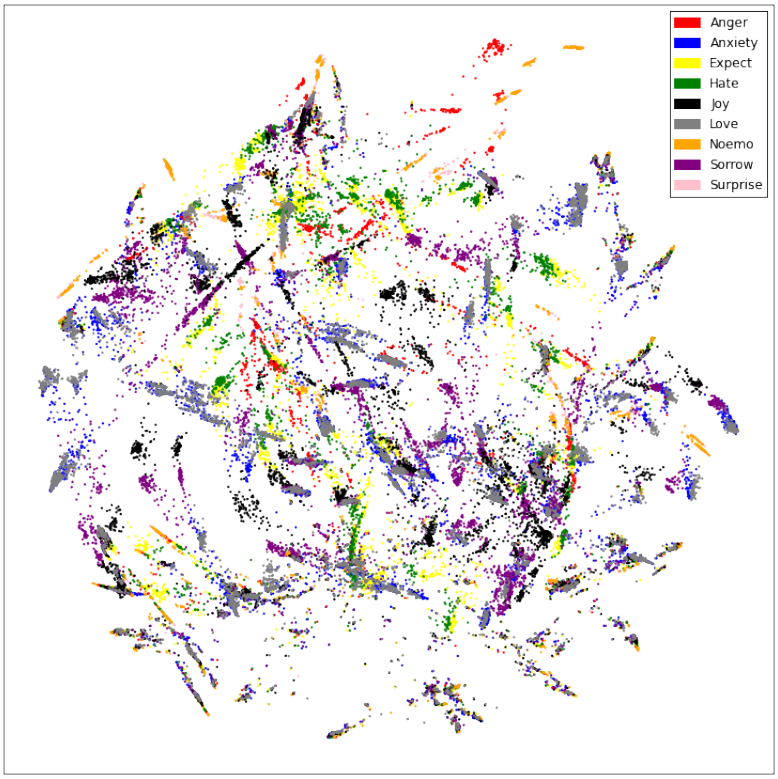

Figure 4.2: Visualization of Ren_CECps in SeTF · IDF

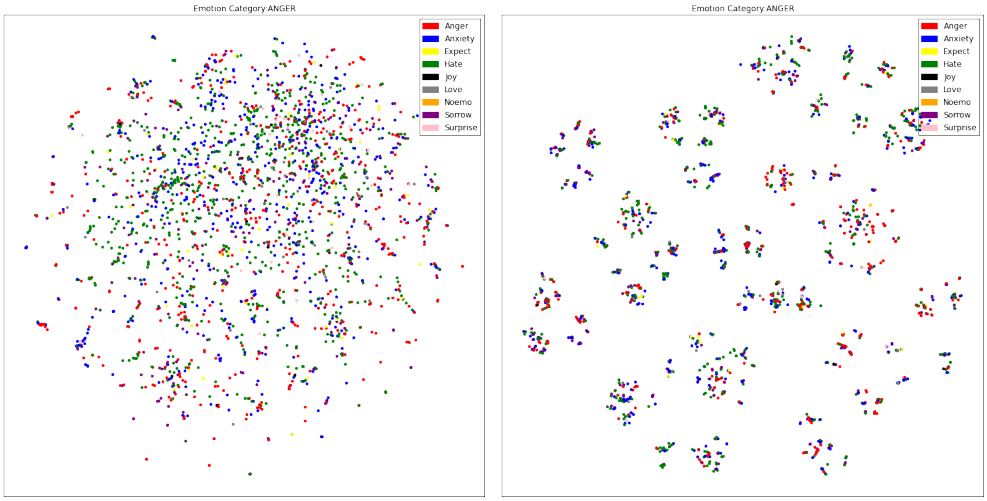

Figure 4.3: Visualization of Ren_CECps in traditional TF·IDF(left) and SeTF · IDF(right) for category "Anger"

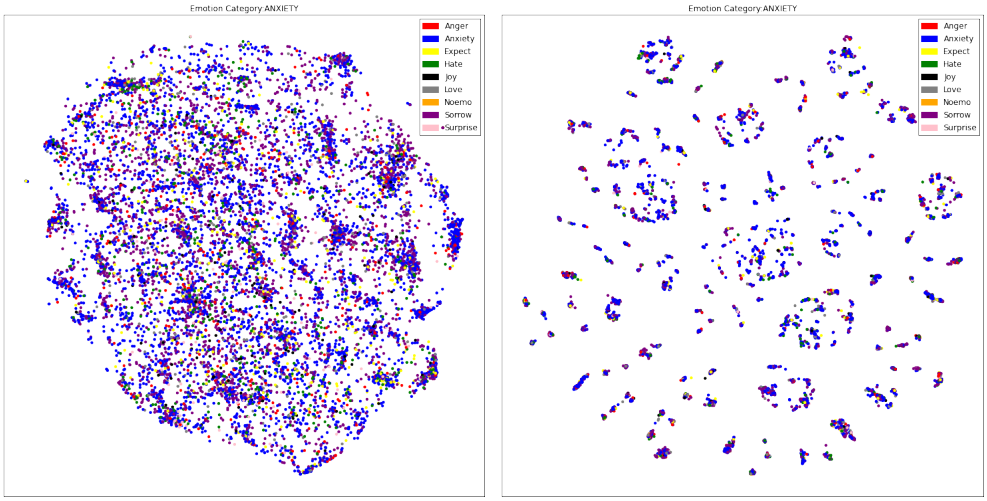

Figure 4.4: Visualization of Ren_CECps in traditional TF·IDF(left) and SeTF · IDF(right) for category "Anxiety"

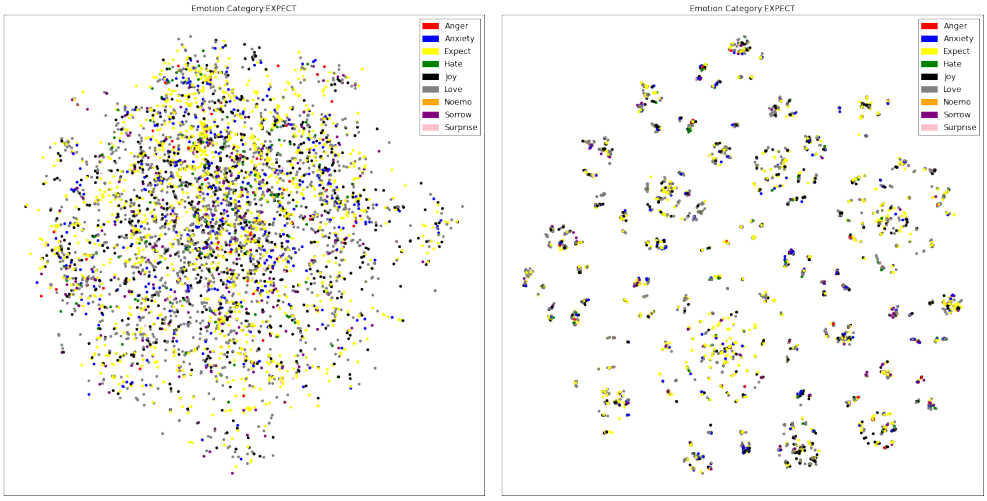

Figure 4.5: Visualization of Ren_CECps in traditional TF·IDF(left) and SeTF · IDF(right) for category "Expect"

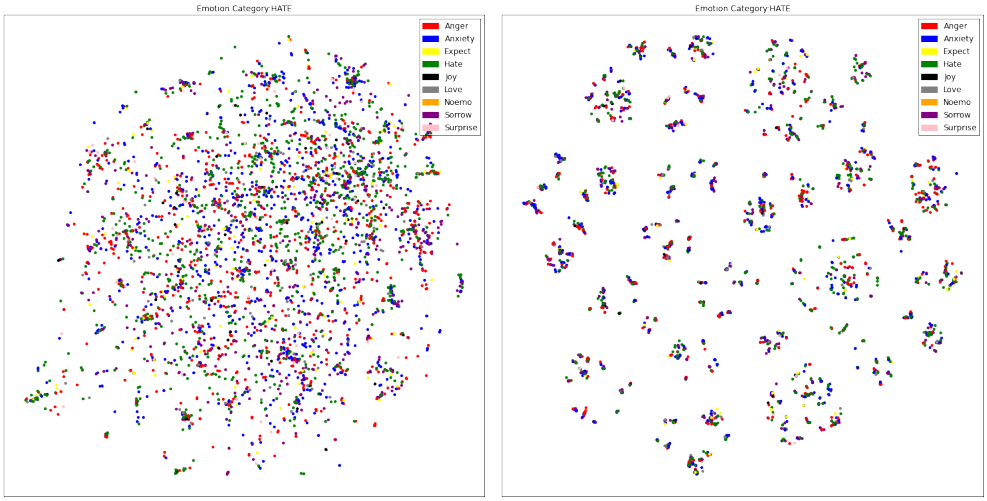

Figure 4.6: Visualization of Ren_CECps in traditional TF·IDF(left) and SeTF · IDF(right) for category "Hate"

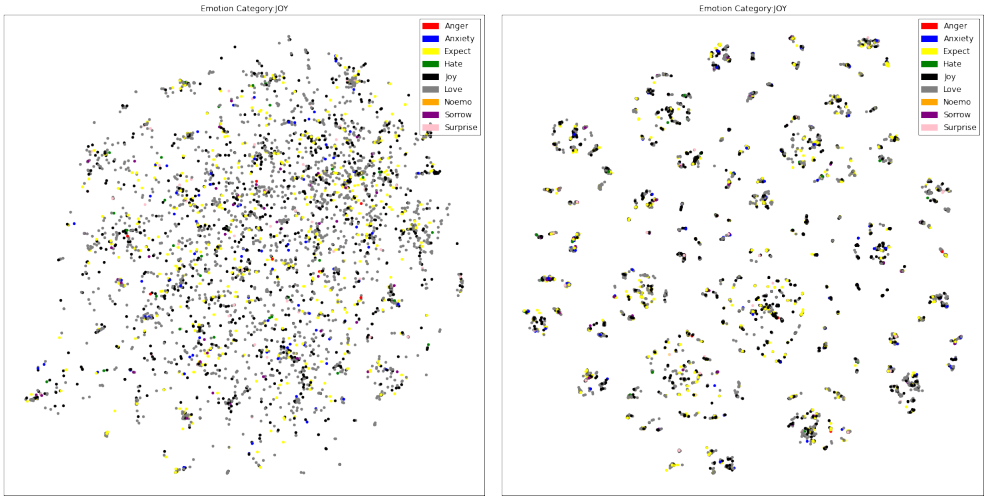

Figure 4.7: Visualization of Ren_CECps in traditional TF·IDF(left) and SeTF · IDF(right) for category "Joy"

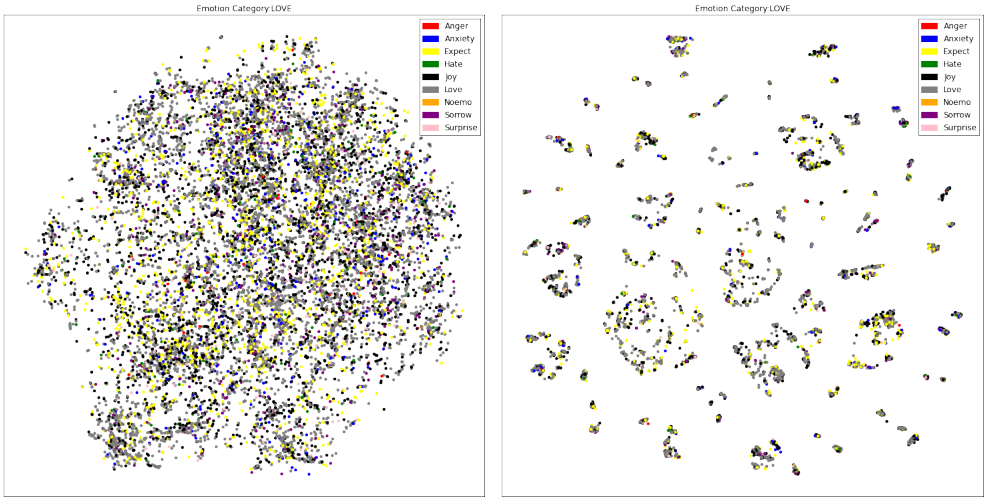

Figure 4.8: Visualization of Ren_CECps in traditional TF·IDF(left) and SeTF · IDF(right) for category ”Love”

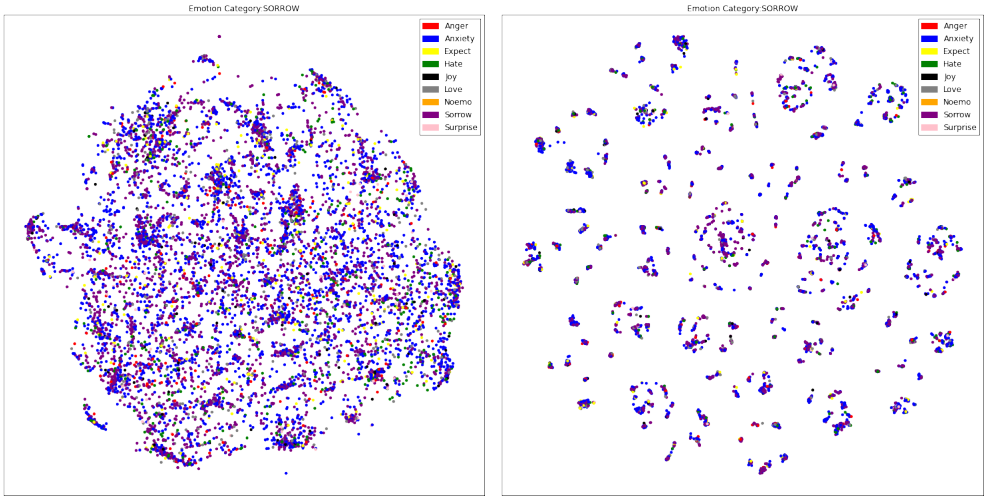

Figure 4.9: Visualization of Ren_CECps in traditional TF·IDF(left) and SeTF · IDF(right) for category ”Sorrow”

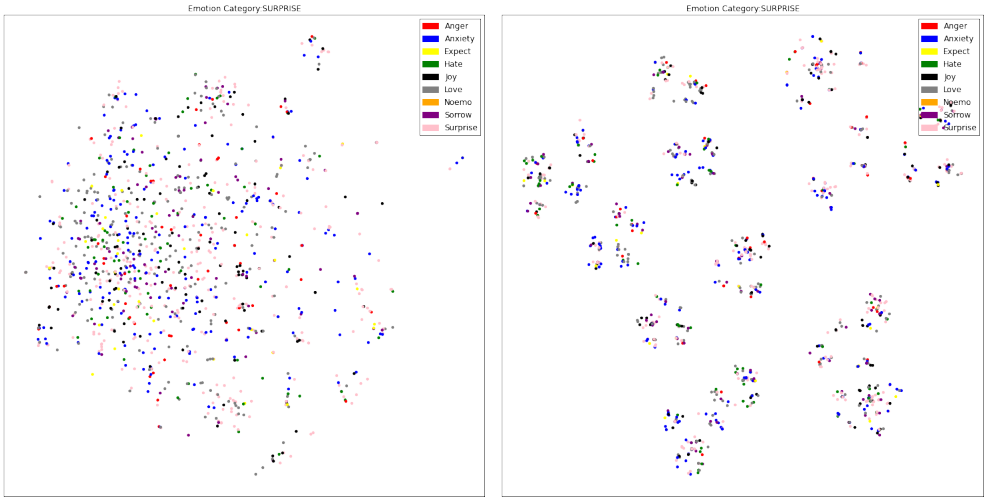

Figure 4.10: Visualization of Ren_CECps in traditional TF·IDF(left) and SeTF · IDF(right) for category ”Surprise”

Chapter 5

Emotion Computing Based on Distance Features and Deep Neural Network

5.1 Word Mover's Distance Features for Emotion Computing

The visualization graphs from Fig.4.1 to Fig.4.10 show the remarkable progress derived from the changed distances among emotional points. It seems like a lot of words walk away from each other and which makes the sentences heavy fog in the left being blown over into air circulation. This inspires us if we can let the words "walking" in the algorithm, maybe we will get better results in classification. Following the desire, we focused on transportation problem in NLP, and found word mover's distance algorithm.

Word Mover's Distance The word mover's distance(WMD)[50] is a good distance measure came from earth mover's distance(EMD)[97]. The EMD problem can be solved as transport problem. As in WMD, the distance between two text documents A and B is the minimum cumulative distance that words from document A need to travel to match exactly the point cloud of document B[50]. The transportation matrix between documents

A and B can be described as formula 5.1 below:

$$\begin{array}{cc} & \begin{array}{ccccccc} \mathbf{word}_1 & \cdots & \mathbf{word}_i & \cdots & \mathbf{word}_n \end{array} \\ & \begin{array}{ccccccc} d_1 & \cdots & d_i & \cdots & d_n \end{array} \\ \begin{array}{c} \mathbf{word}'_1 \\ \vdots \\ \mathbf{word}'_j \\ \vdots \\ \mathbf{word}'_m \end{array} & \begin{array}{c} d'_1 \\ \vdots \\ d'_j \\ \vdots \\ d'_m \end{array} \left[\begin{array}{ccccc} \omega_{1,1} & \cdots & \omega_{1,i} & \cdots & \omega_{1,n} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \omega_{j,1} & \cdots & \omega_{j,i} & \cdots & \omega_{j,n} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \omega_{m,1} & \cdots & \omega_{m,i} & \cdots & \omega_{m,n} \end{array} \right] \end{array} \tag{5.1}$$

Where, $\{\mathbf{word}_i\}$ and $\{\mathbf{word}'_j\}$ represent the words in document A and document B respectively. $\{d_i\}$ and $\{d'_j\}$ mean the term frequencies of the corresponding words. $\omega_{j,i}$ is the distance between $\mathbf{word}'_j$ and $\mathbf{word}_i$, especially the distance is undirected.

To measure the distances of two words, every words are represented as vectors provided by trained *word2vec* embedding matrix $\mathbf{V}$, and the distances can be calculated by Euclidean distance in formula 5.2:

$$\omega_{j,i} = \parallel \mathbf{V}_j, \mathbf{V}_i \parallel_2, \mathbf{V}_{j,i} \in \mathbf{V} \tag{5.2}$$

Let $\mathbf{T}_{ij}$ where $i \in [1, n], j \in [1, m]$ be the number of $\mathbf{word}_i$ in document A which transports into $\mathbf{word}'_j$ of document B. In this way, $\sum_{j=1}^{m} \sum_{i=1}^{n} \mathbf{T}_{ij}$ denotes the total numbers of words in document A transporting into the words of document B. On the contrary, $\sum_{i=1}^{n} \sum_{j=1}^{m} \mathbf{T}_{ji}$ means the reverse direction of the transportation.

Thus, the WMD of documents distance measurement can be described as an optimiza-

tion problem in formula 5.3, and the minimum result is the distance of two documents.

$$\begin{aligned} \min \quad & \sum_{j=1}^{m}\sum_{i=1}^{n}\mathbf{T}_{ij}\omega_{j,i} \\ \text{subject to:} \quad & \sum_{j=1}^{m}\sum_{i=1}^{n}\mathbf{T}_{ij} = \sum_{j=0}^{m} d'_j, \\ & \sum_{i=1}^{n}\sum_{j=1}^{m}\mathbf{T}_{ji} = \sum_{i=0}^{n} d_i, \\ & \mathbf{T}_{ij} \geq 0. \end{aligned} \tag{5.3}$$

Here is an example of calculating the similarities of two target sentences S1, S3 with a standard sentence S2 in WMD and TF · IDF, the sentences are all tokenized and split with blank space :

S1: "风和日丽.(English: Sunny Days.)"

S2: "天气　很　好 .(English: It's a good day.)"

S3: "今天　下　雨.(English: It's raining today.)"

We use the *cosine* function to calculate the vectors represented in TF · IDF as the similarity measurement and the WMD are calculated following the "word mover's distance in python"[1] published by vene&Matt Kusner[50]. Assuming $sim()$ formula as the similarity between two sentences, we can describe the results below:

$WMD : sim(S1, S2) = 0.75, sim(S3, S2) = 0.82.$

$TF \cdot IDF : sim(S1, S2) = 1.0, sim(S3, S2) = 1.0.$

In TF · IDF, S1 and S3 get the same similarity results. In WMD, S1 gets a lower result compared with S3, this means S3 is farther away from s2 than S1 from S2. Or can be said that S2 is more similar to S1 than S3, and this matches the ground truth.

As the example shows, WMD has an ability to measure the semantic difference between sentences. Thus we can use several selected sentences as a core dataset, the samples in the entire corpus can be represented by its similarities with all of the sentences in the core dataset.

[1] http://vene.ro/blog/word-movers-distance-in-python.html

Fast computing

The EMD has a best average time complexity of $O(N^3 \log N)$[82], where N denotes the vocabulary length. This means the lower scale of words, the faster the computing will be. We continue to use $\mathbf{V}$ as the word embedding, the pseudo-code of calculating WMD of documents used in[50] can be show in Algorithm 2, a pseudo-code of fast WMD for comparing is presented in Algorithm 3.

Algorithm 2 WMD

1. **for** corpus D **do**
2. T-D matrix $\mathbf{TD} \leftarrow D$
3. **end for**
4. **for** words in **TD do**
5. distance matrix $\mathbf{M} \leftarrow \mathbf{TD}, \mathbf{V}$
6. **end for**
7. **loop** d_i, d_j in corpus D
8. **return** emd(**TD**[i],**TD**[j],**M**)
9. **end loop**

Algorithm 3 fast WMD

1. **loop** i, j in index
2. **for** d_i, d_j in corpus D **do**
3. T-D matrix $\mathbf{TD'} \leftarrow [d_i, d_j]$
4. **end for**
5. **for** words in $\mathbf{TD'}$ **do**
6. distance matrix $\mathbf{M'} \leftarrow \mathbf{TD'}, \mathbf{V}$
7. **end for**
8. **return** emd($\mathbf{TD'}$[0],$\mathbf{TD'}$[1],$\mathbf{M'}$)
9. **end loop**

As shown above, the matrix **TD** needed for WMD in algorithm 2 is exported from the whole documents of corpus, this is a fully vocabulary length matrix with dimension of tens of thousands in Chinese or hundreds of thousands in English. In algorithm 3, we export the matrix **TD**' only from two documents which need to be calculated. This makes

the dimension of **TD**' far less than **TD**, and restricts the dimension into one hundred. For each computation of WMD, though the T-D matrix $\mathbf{TD}'$ and distance matrix $\mathbf{M}'$ are both needed to be recomputed within every loop process, the fast WMD still has a lower computational complexity compared with the exponential order complexity of EMD in step 8 of algorithm 2.

To verify the improvement, we make three groups of experiments. Each group is a ten times' computing based on 10 pairs of sentences which are selected randomly from Ren_CECps. The comparison of time-consuming is illustrated in Table 5.1.

Table 5.1: The comparison of time-consuming in WMD and fast WMD

Groups	**Case 1**	**Case 2**	**Case 3**
	per 10 times(s)		
WMD	632.545	646.700	646.237
fast WMD	0.047	0.042	0.031
rate	**16000 times**		

Parallelization Though the fast WMD algorithm has a 16000 times improvement of computation efficiency compared with WMD algorithm, it's still too slow for our experiments, so we parallelize the fast WMD model using 10-12 processes in 8 servers when verifying the ability of distance features in emotion computing experiments. To be consist, the words of WMD written in the rest part of this thesis are all in the meaning of the fast WMD algorithm without special explanation.

5.2 Multi-label Computing Using Deep Neural Network Based on Distance Features

The WMD features can measure the difference among samples points, the hidden layers in deep neural network can also learn complex features of a text, and for multi-label annotated emotional corpus, the hidden emotions can be recognized through the transmission cells. Thus, we want to verify whether a deep neural network can handle the multi-label emotion classification using WMD features as input or not.

5.2.1 Multi-layer Dense Neural Network

In this section, we construct a three-layer dense neural network to recognize multi-label emotional corpus. As can be saw in Fig.5.1. The input Word Mover's Distance(WMD) algorithm calculated feature vector of the emotional sentences is a 1-D vector.

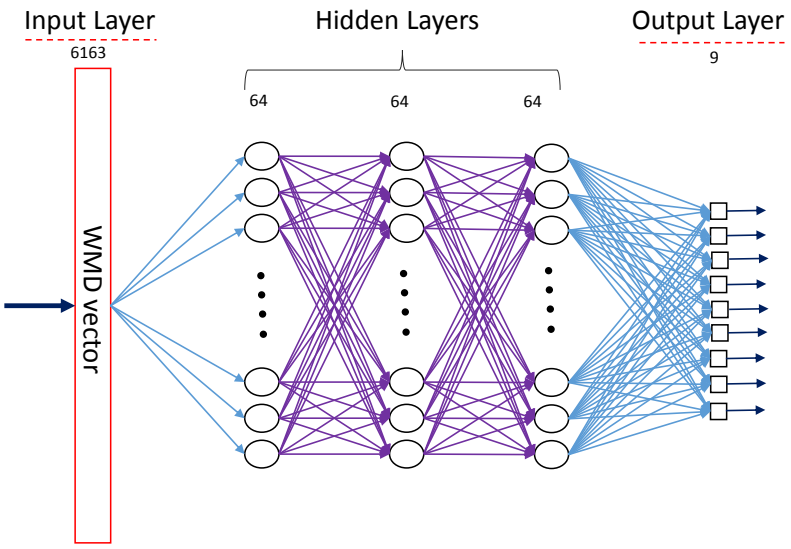

Figure 5.1: The construction of multi-layer dense neural network

In this paper, we randomly selected 6163 sentences as seed corpus calculated in WMD algorithm. As can be saw in Fig. 5.1, the "Input Layer" is the 6163 dimensional WMD vector. We use rectified linear function(relu) as activation function in function 5.4 below.

$$f(x) = max(x, 0) \tag{5.4}$$

where, x means feature.

The next part of DNN is a three-layer dense network, which all has 64 cells with activation function of rectified linear function. This hidden layers are all transmitted without weight specified. The final is a nine independent 'sigmoid' output layer. The nine outputs represent 'anger', 'anxiety', 'expect', 'hate', 'joy', 'love', 'neutral', 'sorrow', 'surprise' respectively.

To optimize the network, we make use of rmsprop as the optimizer. rmsprop is a mini-batch version of rprop, and can be described in function 5.5 below:

$$
\begin{aligned}
E[g^2]_t &= 0.9E[g^2]_{t-1} + 0.1g_t^2 \\
\theta_{t+1} &= \theta_t - \frac{\eta}{\sqrt{E[g^2]_t + \epsilon}} g_t
\end{aligned}
\tag{5.5}
$$

where,$E[g^2]_t$ means the running average at time step t, g_t means the gradient of the objective function at time step t, θ_t means the updated parameters at step t.

For a better network, we choose binary cross entropy ("binary_crossentropy")as loss function. The "binary_crossentropy" loss is a special case of multinomial cross-entropy loss, as shows in function 5.6:

$$
J(\theta) = -\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{m} y_{ij} log(p_{ij}), \qquad m = 2 \tag{5.6}
$$

where,i means the samples and j represents the sample label, $p_{ij} \in (0, 1)$.

5.2.2 Multi-label Emotion Recognition

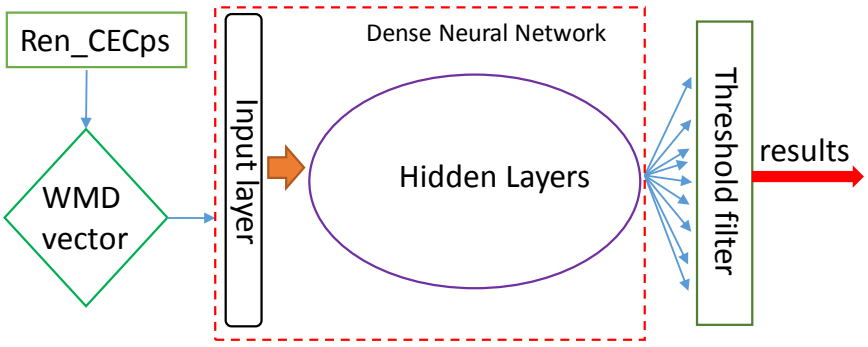

Figure 5.2: The flowchart of multi-label emotion computing

In our experiments, the annotated emotional labels will be represented as a nine dimensional vector $(e_1, e_2, ..., e_i, ..., e_9)$, where $e_i = 1$ if the corresponding emotion category exited or $e_i = 0$ if the corresponding emotion category not annotated. Our goal is to train the model to recognized the nine emotions in sentence level.

Fig. 5.2 shows the flowchart of computing the multi-label emotional corpus. First we learn the feature vector of train and test data, the "WMD vector" part is the distance

feature vector conversion process. The converted feature vector will next be transmit into dense neural network. Through three layers network, we get a nine dimension vector which means the truth probabilities of the nine predicted labels based on the ground truth .

The next is a threshold filter part, which can convert the probabilities achieved from the former section into 1 or 0 labels. Thus we can use this label to measure our model.

Chapter 6

Related Task

In this chapter, we will introduce two tasks: the Temporalia in NTCIR and eRisk in CLEF. Through two tasks, some models used in the thesis or ideas inspired for the thesis are achieved.

6.1 Temporalia in NTCIR

Successful search engines are supposed to consider temporal aspects of information because of the crucial role in estimating information relevance of time [42]. With successful solutions, search engines could then treat temporal queries accordingly to their underlying temporal classes[43].And an analysis [44] shows that users searching for fresh information also seek for information in past as well as future. From this way, we can assuming that the emotions in the text will also change by the time. Temporal related emotion computing will can enhance the emotion recognition in the long time memory sequence.

6.1.1 Annotation Corpus

In Temporal Intent Disambiguation (TID) Subtask, 300 different queries in temporal attribute need to be calculated. Our corpus for dry run and formal run is called SougouCA which was published by Sougou Labs[109].

According to the contents of Temporal and named entities pairs news, we can get annotated sentences using "TemporaliaChTagger" [24, 31]. The news contents in SougouCA were annotated on sentences level, and using <SE> </SE> representing the sentence's

begin and end respectively.

The sentence was annotated mainly into two categories: one is time represented as <T>***</T> and the other one is named entity represented as <E>***</E>. For category <T>,its type is "DATE", and which has four values, named "PRESENT", "PAST", "FUTURE" and a specific date format as "year-month-day". Four of the samples can be shew blew:

1. $<Ttype="DATE"value="PRESENT_REF">$现在(English: now)$</T>$
2. $<Ttype="DATE"value="PAST_REF">$近日(English: recently)$</T>$
3. $<Ttype="DATE"value="FUTURE_REF">$不久(English: soon)$</T>$
4. $<Ttype="DATE"value="2011-08-25">$2 0 1 1 年 8 月 2 5 日(English: 2011.08.25)$</T>$

For category <E>, it has five types, named "ORGANIZATION", "GPE", "PERSON", "LOCATION", and "MISC"(by merging all but the four most dominant entity types into one general entity). The five Types can be shew in the following annotation samples:

1. $<Etype="ORGANIZATION">$公安部治安局(English: Ministry of Public Security Bureau)$</E>$
2. $<Etype="GPE">$唐山(English: Tangshan)$</E>$
3. $<Etype="PERSON">$刘绍武(English: Shaowu Liu)$</E>$
4. $<Etype="LOCATION">$欧美(English: Europe and America)$</E>$
5. $<Etype="MISC">$1 5 0 $</E>$

6.1.2 Word2vec Tool

Word2vec is a tool that takes a text corpus as input and produces the word vectors as output [65]. Relying on large scale corpus, word2vec can train vectors for every words by Skip-gram model. The trained vectors denote the relationship of one word with other words [67]. We use this tool to produce the word vectors of SougouCA. Table 6.1 shows

Table 6.1: Ten similarity words of query string ”滑雪”

Word	Similarity	Word	Similarity
登山(English: Climbing)	0.7583161	探险(English: Expedition)	0.51729447
露营 (English: Camping)	0.6478083	柳汉奎 (English: Ryu Hangyu)	0.5031028
冲浪(English: Surfing)	0.54753125	徒步 (English: Hiking)	0.4999488
热气球(English:Fire Ballon)	0.54753125	冰雪 (English: Ice and Snow)	0.497199
观峰 (English: Sightseeing)	0.52005875	水上 (English: Overwater)	0.492894

the ten similarity words of query string ”滑雪(English:Skiing)” of id 037 computing by the model trained by word2vec.

When training the word vectors, the main parameters for using word2vec are ”-skip-gram -size 200 -windows 5”. For every query string $S_{query} = (w_1, w_2, \cdots, w_i, \cdots, w_n)$, in which $w_i$means the word in query string. In our method, the feature vector for query string V_{query} can be represented as formula(1):

$$V_{query} = \sum_{j=1}^{n} v_{word}(j) \tag{6.1}$$

$v_{word}(j)$ means the vector of the word in query string. n means the number of words in query string.

6.2 Our Experiment System for Task

Followed the algorithm[58], for the query string S_{query} in task, using word2vec tool to train word vectors about SougouCA news corpus. For the sentences $S_{<SE>}$ in SougouCA annotated by TemporaliaChTagger, defined the feature vectors of query and training sentences as formula 6.2 and 6.3:

$$V_{query} = \sum_{j=1}^{n} v_{word}(j) \tag{6.2}$$

$$V_{<SE>} = \sum_{k=1}^{m} v_{<SE>}(m) \tag{6.3}$$

For every query, computing the similarity of $V_{<SE>}$ and V_{query} using *cos* function. Selecting the total ten highest similarity sentences in SougouCA, and using tag ”<T,type= ”date”> ” to match the query timeline to get the final temporal probability of four time categories. For the query string ”滑雪(English:Skiing)” in id 037, the results calculated by our method are shew in Table 6.2:

Table 6.2: Comparison between temporal Results and grand truth of query string "滑雪(English:Skiing)" in id 037

Item	Past	Recency	Future	Atemporal
Truth	0.1	0.0	0.2	0.7
Results	0.247104	0.274131	0.262548	0.216216

6.2.1 Results

As shows in Table 6.3 and Table 6.4, in the formal run, we proposed a similarity computing method for Temporal information query. In the total 300 queries, we got Averaged Per-Class Absolute Loss(APAL) of 0.271564197983333, and Cosine Similarity(CoS) of 0.607726046083874. For every single temporal categories, if the results of a query string contains a temporal category, we mark one time of the certain temporal category. After calculated all of the 300 queries, we get the precision of four temporal categories shown in Table 6.4. As can be seen in Table 6.4, our method get higher precision of 92.7% in "atemporal" category. Through this task, we can found the effectiveness of using multi-label in searching queries, and get a new view about temporal extraction.

Table 6.3: Precision of four temporal categories

Item	Past	Recency	Future	Atemporal
Precision(%)	52.7	58	39.7	92.7

Table 6.4: Results of TID Task

Item	APAL	CoS
Averaged Value	0.271564197983333	0.607726046083874

6.3 eRisk in CLEF

eRisk is the abbreviation of early risk prediction on the Internet. This task war hosted in CLEF for years. The eRisk 2018 contains two sub-tasks, one is Task 1: Early Detection of Signs of Depression, and the other one is Task 2: Early Detection of Signs of Anorexia. The tasks are both running on the contents crawled from social networks with temporal tags.

Keywords model Marked as TUA1B in Task 2 and TUA1C in Task 1 is a simple model in which using some emotional words as targets to measure the authors' situation.

Research in [20] shows pro-anorexia community uses microblogging platform to share image-rich graphic and "triggering" content around internalization of thin body ideals. According to this empirical description, we select the measurement keyword of anorexia as "body", and the strategy for measurement is if the contents of specific chunks contain the keyword, we will give the conclusion for this sample. The same way for depression detection, we give "depression" as the keyword.

TF · IDF with SVM Marked as TUA1D in Task 1 and TUA1A in Task 2. This is a traditional method to assess the situation of mental health. We train the SVM model using a linear kernel API of sklearn[81] upon the full chunks contents. All of the texts in the chunks are processed into feature vectors using TF · IDF package in sklearn. The TF · IDF scores of every post is normalized under l2 function. The same strategy used above: samples detected will not consider in next chunks.

CNN+LSTM model Marked as TUA1A and TUA1B in Task 1, TUA1C in Task 2. We construct a CNN+LSTM based deep neural network as the detection models using keras[16] with TensorFlow[1] as backend.

In this model, the contents of every chunk are preprocessed into one-hot features with an index value of the corresponding word in vocabulary instead of the word itself. Before feeding into the network, adding a padding process to format the length, the "maxlen" is selected as 2000. Other hyperparameters chosen for this model are as follows: "input_length" of Embedding is set to the length of vocabulary, Task 1 is 429700 and Task 2 is 156593, "embedding_size" is 128; The dropout factor is 0.25; For convolution layer, the "filters" is 64, setting "kernel_size" to 5, "padding" with "valid", "strides" with 1 and "activation" is "relu"; Giving MaxPooling the parameter of 4 as "pool_size"; For LSTM layer, the output dimension is 70; The following Dense layer is a one cell classification layer with "Activation" being "sigmoid". For the compiling, choosing "binary_crossentropy" as "loss" and "adam" for "optimizer". The "metrics" is the default setting as "accuracy". Epoches is added as 20 to maximize the accuracy and minimize the loss of training data. In our experiments, the final loss in Task 1 is 0.000108512764422, in Task 2 is 0.00640664884888 respectively and accuracy are both 1.0.

6.4 Results

6.4.1 Data prepossessing

The crawled contents for two sub-tasks are formatted in XML files. Data sets are divided into ten chunks under posting timeline. For Task 1, the data sets contain training and testing sets of 2017 and this year's data for testing. In Task 2, the files only contain training data and testing data, cause this sub-task is a new task in eRisk this year. Contents for every IDs are organized with five tags: one root tag named $< WRITING >$ means one formatted posts, four child tags named $< TITLE >$, $< DATA >$, $< INFO >$ and $< TEXT >$ orderly.

$< TITLE >$ tag gives the title of this post, $< DATA >$ tag marks the post data, $< INFO >$ tag reminds which platforms the posts are crawled and the final $< TEXT >$ tag contains the contents people write-in. As $< TEXT >$ tag and $< TITLE >$ tag may both have no contents in one $< WRITING >$, to gain more textual information, during processing, we combine both $TITLE$ and $< TEXT >$ contents together, and extend the texts to one sentence. Although most of the IDs have a lot of $< WRITING >$ tags with efficient contents in one chunk, there are still IDs with none content in both $< TEXT >$ tag and $< TITLE >$ tag of one chunk and cannot extract useful information except temporal information (not used in this paper). At this moment, the chunk with none content can only contain the texts from former chunks. All of the contents of IDs are extracted into ten chunks, in which chunk1 only contains the current texts of this chunk, the next chunk can contain the contents of former one and contents contained in current chunk, following this way, we get split contents for chunk1, chunk2,$\cdots$,chunk10. Specially, for Task1, the training and testing data sets of 2017 are combined into one data set as new training corpus.

For the risk detection using keyword model, the $< TEXT >$ tag and $< TITLE >$ tag combined sentences of IDs will be very chunk by chunk without summing them together.

6.4.2 Evaluation results

Employing the models mentioned above with the extracted data, we submit the chunk 10th results of Task1 and Task 2 respectively. Table 6.5 and Table 6.6 show the results of

Table 6.5: Results of four models in Task 1

Models	$ERDE_5$	$ERDE_{50}$	F1	P	R
Kerwords model(TUA1C)	10.86%	9.51%	**0.47**	**0.35**	**0.71**
TF · IDF with SVM(TUA1D)	0	0	0.00	0.00	0.00
CNN+LSTM(TUA1A)	10.19%	9.70%	0.29	0.31	0.27
CNN+LSTM(TUA1B)	10.40%	9.54%	0.27	0.25	0.28

Table 6.6: Results of three models in Task 2

Models	$ERDE_5$	$ERDE_{50}$	F1	P	R
Kerwords model(TUA1B)	19.90%	19.27%	0.25	0.15	**0.76**
TF · IDF with SVM(TUA1A)	0	0	0.00	0.00	0.00
CNN+LSTM(TUA1C)	13.53%	12.57%	**0.36**	**0.42**	0.32

Task 1 and Task 2 respectively.

Our final chunk results of three models can be checked in Table 6.5 and Table 6.6. In Task1, the keywords model gets the best F1, precision and recall scores of 0.47, 0.35 and 0.71, two CNN+LSTM models get almost the same results. The two CNN+LSTM models are just two times running feeds with the model. The recall keywords model gets is one of the top recalls in the 11 teams, in which the best is 0.95. In Task 2, we employ three models, the keywords model only gets the best recall of 0.76, CNN+LSTM model get the best F1 and precision scores of 0.36 and 0.42.

TF · IDF with SVM gets both 0 value in Task 1 and Task 2, this makes us confused. This model with l2 normalization of TF · IDF features predict all the IDs as label "2", no risk ID detected in the tenth chunk. Those results make the model unbelievable, thus gets the 0 evaluation results finally.

Table 6.7: Ten chunks results of keywords model in Task 1

Evaluations	chunk1	chunk2	chunk3	chunk4	chunk5	chunk6	chunk7	chunk8	chunk9	chunk10
$ERDE_5$	9.87%	9.95 %	9.96 %	10.03%	10.17%	10.23%	10.29 %	10.35%	10.52%	10.55%
$ERDE_{50}$	9.63%	9.64 %	9.19 %	8.87%	8.90%	8.70%	8.58 %	8.40%	8.47%	8.02%
F1	0.31	0.37	0.41	0.43	0.42	0.44	0.46	0.46	0.45	0.47
P	0.47	0.44	0.44	0.42	0.38	0.38	0.38	0.37	0.34	0.35
R	0.23	0.32	0.39	0.44	0.48	0.53	0.58	0.61	0.65	0.71

As we only submit the tenth chunk results, for a fully analysis of the models, we calculate the ten chunks results of $ERDE_5$, $ERDE_{50}$, F1, P and R for keywords model

in Task 1 and task2, which shows in Table 6.7 and Table 6.8 separately.

Table 6.8: Ten chunks results of keywords model in Task 2

Evaluations	chunk1	chunk2	chunk3	chunk4	chunk5	chunk6	chunk7	chunk8	chunk9	chunk10
$ERDE_5$	15.78%	16.90%	17.50%	17.82%	18.10%	18.62%	19.06 %	19.26%	19.58%	19.90%
$ERDE_{50}$	15.78%	16.88%	15.71%	15.32%	15.60%	16.03%	16.25 %	16.29%	16.22%	15.98%
F1	0.27	0.28	0.27	0.27	0.27	0.26	0.26	0.25	0.25	0.25
P	0.20	0.18	0.18	0.17	0.17	0.16	0.16	0.15	0.15	0.15
R	0.44	0.56	0.61	0.63	0.66	0.68	0.71	0.71	0.73	0.76

In Task 1, checking Table 6.5 and Table 6.7, we can find keywords model get the best F1 scores from the first chunk, while in Task 2, comparison the Table 6.6 and Table 6.8, The keywords model perform worse than in Task 1, though its recall still perform good than CNN+LSTM method. Considering CNN+LSTM model only, with less negative samples CNN+LSTM model gets higher results in Task 2 compared with the two years data of Task 1. This maybe the small scale of the data sets cannot fully train the deep neural network, with more negative data, the model may learn more unbalanced information.

For keywords model, the keywords used are ”depression” and ”body” respectively for Task 1 and Task 2. They both get the best recall scores, While ”depression” gets better results than ”body” in F1 and Precision. These may indicate that in depression risk signs, people are often using the obvious words like ”depression” to express themselves, while in anorexia situation, the disorder of eating may not almost focus on ”body”, there are maybe other hidden words existing. In other words, keywords based emotion detection for some mental background may work well without temporal informations.

Chapter 7

Emotion Trigger System for Humanoid Robot Interaction

7.1 Actroid REN-XIN

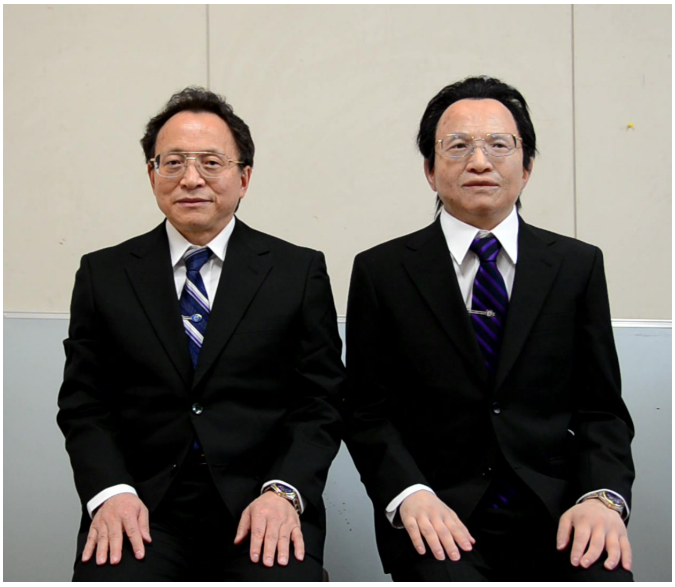

Figure 7.1: Prof. Ren(left) and his avatar robot REN-XIN(right)

Actroid REN-XIN is a humanoid robot developed by kokoro Company Ltd. based on Prof. Ren with almost the same clothes and even the same spectacles as Prof. Ren has, showing in Fig.7.1. The two eyes of REN-XIN are embedded in cameras respectively, and to adjust the myopic lens, the two cameras are also modified with focus.

REN-XIN has 12 degrees of freedom, in which seven for face control, three for head

and the last two for upper body. Below are the detailed introductions for every degrees and their control codes:

- eyebrow up or down;
  - scale, number ranges from 0 to 255, where 0 means the lowest position of eyebrow, 255 means the highest position.
- cheek pulling;
  - scale, number ranges from 0 to 255, where 0 means no control on cheek, at this number, the apples of cheek are not showing in the face, 255 means the highest position to pull the cheek, one this highest position, the apples of cheek can be seen clearly.
- eyelids closed or opening;
  - scale, number ranges from 0 to 255, where 0 means the two eyelids are closed, 255 means opening the eyelids completely.
- eyes turning right or left;
  - scale, number ranges from 0 to 255, where 0 means the most right position of eyes, 255 means the most left position of eyes.
- eyes moving up or down;
  - scale, number ranges from 0 to 255, where 0 means the lowest position to move eyes down, 255 means the highest position to move the eyes up.
- mouth closed or opening;
  - scale, number ranges from 0 to 255, where 0 means closing mouth, 255 means opening mouth completely.
- mouth opening roundly or not;
  - scale, number ranges from 0 to 255, where 0 means no control operation, 255 means opening the mouth roundly at the biggest parameter.

- left-side wry head
  - scale, number ranges from 0 to 255, where 0 means no control operation, it's the normal position, 255 means wrying the head to the most left position.
- right-side wry head
  - scale, number ranges from 0 to 255, where 0 means no control operation, it's the normal position, 255 means wrying the head to the most right position.
- head rotation
  - scale, number ranges from 0 to 255, where 0 means rotating the head to the most right angle, 255 means rotating the head to the most left angle.
- shrug or not
  - scale, number ranges from 0 to 255, where 0 means no shrug, 255 means shrugging the shoulder to the highest position.
- body leaning forward or backward
  - scale, number ranges from 0 to 255, where 0 means leaning the body to the most backward angle, 255 means leaning the body to the most forward angle.

⋆ Cause the servos are different from their initial position, not all of them start from the zero value, and in some spacial situation, the maximum value 255 can not be achieved.

7.2 Emotion Enhanced Interaction for REN-XIN

As mentioned above, almost all of the humanoid robot have to encode the face actions manually. This makes it impossible to control the robot in real-time interaction with human beyond the scripts. To give a solution for the unknown situations, we embed the fast WMD model to recognize the emotion categories as the emotional action trigger during the interaction with REN-XIN. In this way, we only need to encode several emotional actions for our humanoid robot REN-XIN. As the fast WMD model was trained based

on Ren_CECps considered with neutral emotion, to coordinate with the emotional action trigger, we only need to make eight emotional actions and another one neutral face.

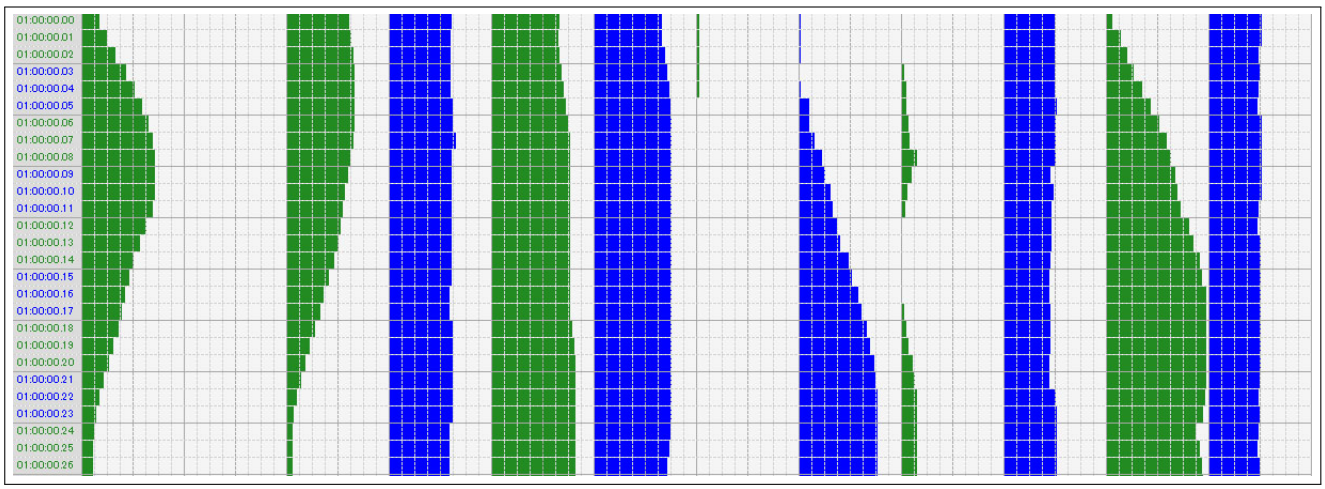

Figure 7.2: The sample fragment of action encoder for REN-XIN

Fig.7.2 shows the sample fragment of action encoder with GUI for REN-XIN. And Fig7.3 is the scale result of the sample fragment in Fig7.2. Every row represents one movement per 1/60s. In Fig.7.2, there are 98 columns per row, and each 8 column represent the control code range of servos. There are 12 ranges totally corresponding to the 12 degrees mentioned above in Section 7.1.

```
00:00:00.00, 45,0,157,157,170,170,8,5,0,127,16,133,
00:00:00.01, 64,0,162,154,168,170,8,5,0,127,37,133,
00:00:00.02, 85,0,168,154,170,178,8,5,0,127,53,125,
00:00:00.03, 112,0,170,154,176,184,8,0,8,128,69,128,
00:00:00.04, 133,0,170,154,181,189,8,5,13,128,90,125,
00:00:00.05, 152,0,170,160,186,192,2,26,13,133,112,122,
00:00:00.06, 168,0,170,160,192,194,0,26,18,130,133,133,
00:00:00.07, 178,0,168,168,197,194,0,40,21,130,152,133,
00:00:00.08, 184,0,160,157,197,194,0,58,40,130,162,133,
00:00:00.09, 184,0,154,157,197,194,0,66,26,117,173,133,
00:00:00.10, 184,0,146,157,197,194,0,80,16,125,178,133,
00:00:00.11, 178,0,141,157,197,194,0,85,10,120,186,125,
00:00:00.12, 162,0,136,157,197,194,0,96,2,120,208,122,
00:00:00.13, 146,0,130,157,197,194,0,104,2,120,218,130,
00:00:00.14, 130,0,120,157,197,194,0,125,0,117,234,130,
00:00:00.15, 120,0,106,157,197,194,0,133,0,114,240,130,
00:00:00.16, 109,0,93,152,197,194,0,149,0,114,250,130,
00:00:00.17, 101,0,85,152,197,194,0,157,8,117,250,130,
00:00:00.18, 93,0,72,160,202,194,0,170,13,117,250,130,
00:00:00.19, 80,0,58,160,208,194,0,178,18,117,250,130,
00:00:00.20, 69,0,48,160,210,194,0,189,29,114,250,130,
00:00:00.21, 56,0,37,160,210,194,0,192,34,114,250,128,
00:00:00.22, 45,0,26,160,210,194,0,197,40,130,248,125,
00:00:00.23, 37,0,18,160,210,194,0,197,40,133,242,128,
00:00:00.24, 32,0,16,152,210,192,0,197,40,133,224,125,
00:00:00.25, 29,0,16,152,210,189,0,197,40,133,234,122,
00:00:00.26, 29,0,16,152,210,184,0,197,40,133,240,128,
```

Figure 7.3: The exported position value of the sample fragment in Fig.7.2

In this sample, we have 26 movements, some servos are controlled by linear variation

control codes, while some are discretely encoded. The colored histogram means the encoded value of corresponding servo, and the whitespace means 0. In this way, the nine emotional actions can be encoded, and are ready to activate by the emotional triggers.

According to our three time-consuming experiments, the fast WMD need 0.047s, 0.042s and 0.031s to finish a ten times computing separately in Section 5.1. A fully computed input vector has 1800 dimensions, it's a 1800 times' computing based on the seed corpus. In average, we need 7.2 second to get the emotional trigger ready.

7.3 DNN Models for Emotional Triggers

During the running interaction with Actroid REN-XIN, the fast WMD based emotional trigger system needs at least 7s to deal with the response. To make a real time interaction, the seamless user experience is a essential aspect. In this thesis, we propose a CNN+LSTM based network as the emotional trigger. For a full comparison, the CNN and LSTM based only networks are also designed in the experiments.

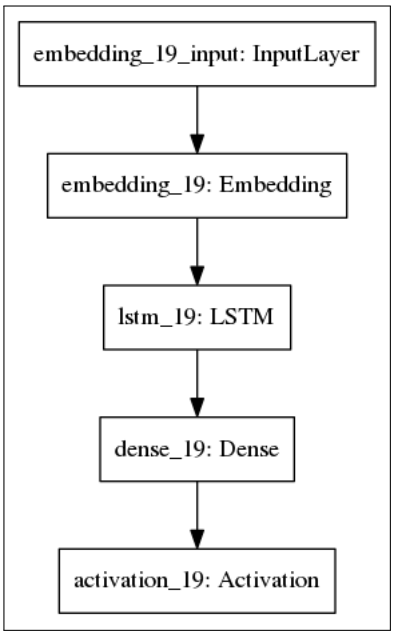

Figure 7.4: LSTM based network

7.3.1 LSTM based structure

The LSTM based network is much light than the two networks below. One Embedding layer with a LSTM layer followed, the final is a Dense layer for output the emotional triggers. Fig.7.4 shows the architecture.

7.3.2 CNN+LSTM based structure

The CNN+LSTM based structure has been proved to achieve excellent performance on sentiment classification[126]. In this part, we construct the traditional layers to combine the speed of CNN and the temporal semantics of LSTM together. As showing in Fig.7.5. Input layer is followed with an embedding layer to train a sentence vector for the next CNN layer.

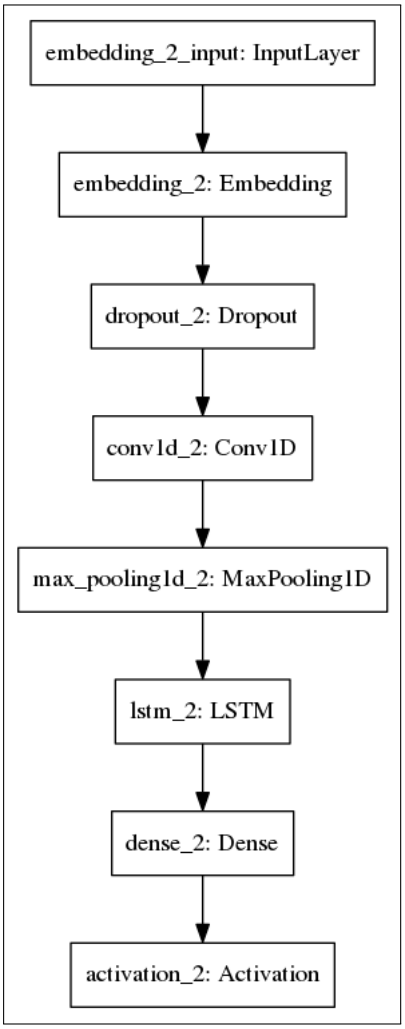

Figure 7.5: CNN+LSTM based network

7.3.3 CNN based structure

The CNN based structure is a simple convolution network in which the ”Flatten” layer takes place of the ”LSTM” layer. The connection layers are described detailedly in Fig.7.6.

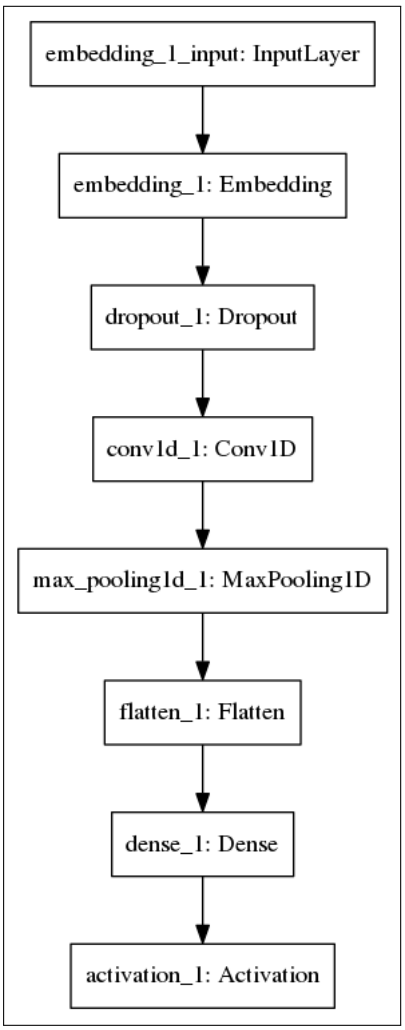

Figure 7.6: CNN based network

Chapter 8

Evaluation and Results

Evaluation measures In this chapter, the evaluation and results based on the former algorithms and models will be presented. All of the precision and recall scores are calculated in micro or macro or the both standards, F1-score :

$$F1 = \frac{2 * precision * recall}{precision + recall} \tag{8.1}$$

where:

$$precision = \begin{cases} Mean(\sum \frac{tp_i}{tp_i + fp_i}) & , \quad if \quad macro \\ \frac{tp}{tp+fp} & , \quad if \quad micro \end{cases}$$

$$recall = \begin{cases} Mean(\sum \frac{tp_i}{tp_i + fn_i}) & , \quad if \quad macro \\ \frac{tp}{tp+fn} & , \quad if \quad micro \end{cases}$$

In which, tp is the number of global true positive, fp is the number of global false positive and fn is the number of global false negatives when using micro standard; tp_i is the number of true positive of category i, fp_i is the number of false positive of category i and fn_i is the number of false negatives of category i when using macro standard. In this paper, both of *precision* and *recall* are calculated in 'macro' and 'micro' model using *metrics* package[1] in sklearn[81].

[1] http://scikit-learn.org/stable/modules/classes.html#module-sklearn.metrics

8.1 Evaluation of Word Mover's Distance Based Features

We evaluate the WMD features in SVM[18] model on Ren_CECps, and regard TF · IDF[99], SeTF · IDF as baseline and enhanced baseline respectively. To be comprehensive, two low dimensional feature representation methods will be evaluated. For a better comparison, an English corpus based experiment is further added.

8.1.1 Dataset and setup

Ren_CECps The corpus is divided into nine single label data sets. Sentences with multi-label will be replicated in every categories. We select 200 sentences from every nine emotion categories randomly as the seed corpus, naturally, the dimension of the WMD features is 1800. Based on the divided corpus, two ways of selecting subsets for the experiments will be executed: one is 50% of the data for training and the rest 50% of the data for testing; the other one is 80% of the data for training and the rest 20% of the data for testing, the selection is random.

20 newsgroups data set We utilized the split "train" and "test" data sets[10] provided by sklearn tools[2]. All of the "headers", "footers" and "quotes" in data sets are removed. The number of the seed documents for the 20 news categories is 100 and the selection is also random from "train" subset.

Word embeddings The word embeddings used in this paper are different with languages. For Chinese, we merged two additional Chinese data sets(sougouCA[3]: A Chinese news corpus published by Sougou Lab[109] and People's Daily data set: We collected 11,355 days' news data from 1980.01.01 to 2016.02.14 through the Internet) into Ren_CECps to train a 200 dimension word embedding using *gensim*[90] which is a free python library containing the approach in [67]. For English, a pre-trained embedding[4] will be utilized for the experiments, which contains 300-dimension vectors for 3 million words and phrases. Both in Chinese and English experiments, the words not in the embeddings will be determined as zero vectors.

[2]http://scikit-learn.org/stable/datasets/twenty_newsgroups. html
[3]https://www.sogou.com/labs/resource/ca.php
[4]https://code.google.com/ archive/p/word2vec/

8.1.2 Results

The experiments are arranged based on 50321 sentences(split in categories) of Ren_CECps and 18846 documents of 20 newsgroup. We use a Linear Support Vector Machine library[5] for our classification experiments. All of the SVM programs are running upon the default configuration.

We will make some classification experiments among TF · IDF, SeTF · IDF, WMD and a sentence embedding method which is one of the stat-of-the-art methods trained by sent2vec[78]. To make a full comparison with the feature's dimensions, some low dimensional feature representation based experiments with TF · IDF and selected seed sub-corpus of the two language corpus will be carried out. For a better comparison for whether word embedding has a more important influence above WMD or not, we add another experiment in which a method combined TF · IDF and word embedding will be experimented. Here are some abbreviations using all through this section:

- **1v1:** The experiments based on the 50% of Ren_CECps for training and the rest 50% of Ren_CECps for testing;
- **4v1:** The experiments based on the 80% of Ren_CECps for training and the rest 20% of Ren_CECps for testing;
- **20 news:** The experiments based on the training and testing data of 20 newsgroup data set;
- **TF · IDF$_{1800}$:** A low dimensional feature representation method, using SVD[6] to reduce TF · IDF feature vectors into 1800 dimensions;
- **TF · IDF$_{2000}$:** The 2000 dimensions feature representation method reduced from TF · IDF feature vectors;
- **Seed_TF · IDF:** Using cosine function to calculate the similarity between target data and seed sub-corpus, in which the final similarities will be the feature vectors of training and testing data. In these experiments, all of the data are initialized by TF · IDF. We assume $(v_1, \cdots, v_i, \cdots, v_n)$ as the TF · IDF represented sub-corpus, in

[5]http://scikit-learn.org/stable/modules/generated/sklearn.svm. LinearSVC.html
[6]http://scikit-learn.org/stable/modules/generated/sklearn.decomposition.TruncatedSVD.html

which v_i is the TF · IDF vector. t_j means the TF · IDF vector of training and testing data. Thus, the final feature vectors of training and testing data can be calculated as this function: $(\cos(v_1, t_j), \cdots, \cos(v_i, t_j), \cdots, \cos(v_n, t_j))$

- **TF · IDF_word2vec:** The enhanced TF · IDF method which uses the word embeddings trained by word2vec as the weights for the corresponding words. The feature vectors used in this experiment are exported by the multiplication of TF · IDF and the embedding matrix.

- **sent2vec:** In this experiment, we use sentences embeddings trained by sent2vec tool[7] as feature vectors. Every documents of 20 newsgroup data set are converted into one line files. The output dimension of sentence embeddings is 700.

Results pre-processing As the results computed by WMD contain the "NaN" data. In order to fit data into SVM, we convert all of the "NaN" data into integer of zero. In discussion section, we will explain the reason.

Table 8.1 shows the results of classification experiments based on the feature representation methods mentioned above, the best and worst results of three experiments are all marked in bold. According to the results, we drew some histogram graphs below. Fig.8.1 shows the comparison graph between 1v1 and 4v1 experiments, and Fig.8.2 shows the results in 20 newsgroup.

In Fig.8.1, we can find that both in 1v1 and 4v1 experiments, the WMD algorithm gets the best classification results, which even higher than the manual emotion separated method of SeTF · IDF over about three percentage points. When compared with the same dimensional features, the WMD shows strong information representation capability, and gets 20% higher score than the low dimension TF · IDF_{1800} and 5% higher than similarity representation method of Seed_TF · IDF.

But in Fig.8.2, the WMD method is knocked off in English news experiments. The 20 newsgroup based experiments get the best results in TF · IDF, and WMD gets nearly the same F1-score with Seed_TF · IDF. Both of the two methods are 5% lower than TF · IDF model. One thing makes us excited is WMD method is still higher than the low dimension

[7]https://github.com/epfml/sent2vec

Table 8.1: The results of experiments on Ren_CECps and 20 newsgroup

Type	Algorithm	Precision	Recall	F1-score
1v1	TF · IDF	0.210819957	0.197094468	0.203726295
	SeTF · IDF	0.355171204	0.236033564	0.283598272
	$TF \cdot IDF_{1800}$	0.116894026	0.115778614	**0.116333646**
	WMD	0.358587638	0.273787523	**0.310501826**
	Seed_TF · IDF	0.284783174	0.227757698	0.253098099
	TF · IDF_word2vec	0.218511086	0.223298753	0.220878979
	sent2vec	0.209975675	0.153755042	0.177520441
4v1	TF · IDF	0.203162567	0.190372868	0.196559888
	SeTF · IDF	0.361914601	0.246556037	0.293300035
	$TF \cdot IDF_{1800}$	0.117098454	0.113943072	**0.115499216**
	WMD	0.338477937	0.300256706	**0.318223762**
	Seed_TF · IDF	0.29824698	0.233749726	0.262088652
	TF · IDF_word2vec	0.238826227	0.231257587	0.234980977
	sent2vec	0.223046830	0.154791461	0.182754080
20 news	TF · IDF	0.688655922	0.680110185	**0.684356377**
	$TF \cdot IDF_{2000}$	0.078399628	0.07408649	**0.07618206**
	WMD	0.701975748	0.598521007	0.646133456
	Seed_TF · IDF	0.654635361	0.647471893	0.651033922
	TF · IDF_word2vec	0.582910280	0.581137014	0.582022296
	sent2vec	0.246903835	0.264207205	0.255262622

$TF \cdot IDF_{2000}$ and gets almost 10 times promotion on F1-score.

We can find the TF · IDF_word2vec method gets the F1-score of 0.2209, 0.235 and 0.582 respectively in Fig.8.1 and Fig.8.2, and are all lower than the results got by the WMD method of 0.3105, 0.3182 and 0.6461. The sent2vec experiments haven't got the best results than other methods, the F1-scores are only better than $TF \cdot IDF_{2000}$ and $TF \cdot IDF_{1800}$.

In Fig.8.1 and Fig.8.2, all of the low dimension feature representation methods reduced from TF · IDF model get the worst results. But selected seed corpus based similarity representation algorithm gets higher results in 1v1 and 4v1 experiments of Chinese corpus than TF · IDF model, gets lower results in 20 newsgroup experiments of English corpus in the contrary. After digging into the feature dimensions of those methods, we found the dimensions of TF · IDF vectors in Chinese and English corpus are 30,000 and 130,000 in integer respectively. The dimensions of WMD method in the two corpus are 1800 and 2000 separately as mentioned before. Computing the rate of dimension reduction of WMD, the Chinese corpus got a reduction rate of 17:1 and English corpus got a reduction rate of 67:1, this may explain why WMD perform better in Chinese corpus than in English

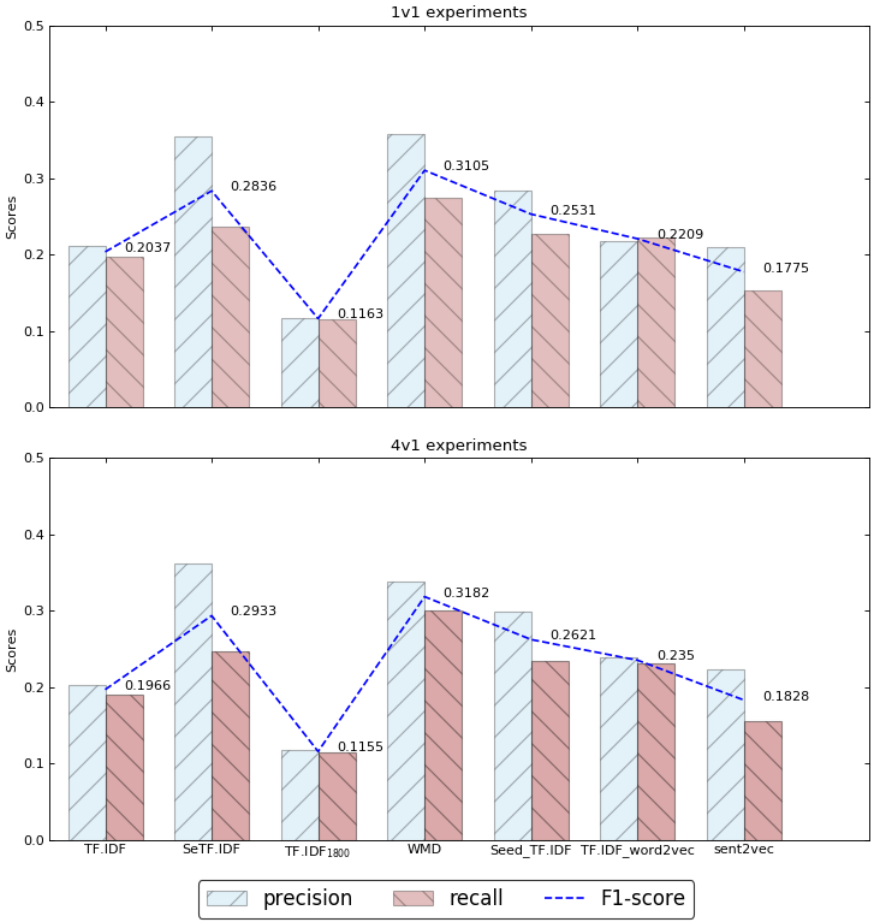

Figure 8.1: The results of 1v1 and 4v1 experiments

corpus. The same situation can be found in the results between TF · IDF and dimension reduced TF · IDF of $\text{TF} \cdot \text{IDF}_{1800}$ and $\text{TF} \cdot \text{IDF}_{2000}$: In 1v1 and 4v1 experiments, the F1-scores of $\text{TF} \cdot \text{IDF}_{1800}$ drop two times compared with TF · IDF(0.22 to 0.11, 0.196 to 0.115), while in 20 newsgroup, F1-scores of $\text{TF} \cdot \text{IDF}_{2000}$ decline nine times compared with TF · IDF(0.68 to 0.07).

8.2 Evaluation of Multi-label Computing Using Deep Neural Network Based on Distance Features

In this part, we use keras[16] to construct the deep neural network. The epochs set up in the model is 50, optimizer is ”rmsprop”, and loss function is ”binary_crossentropy”.

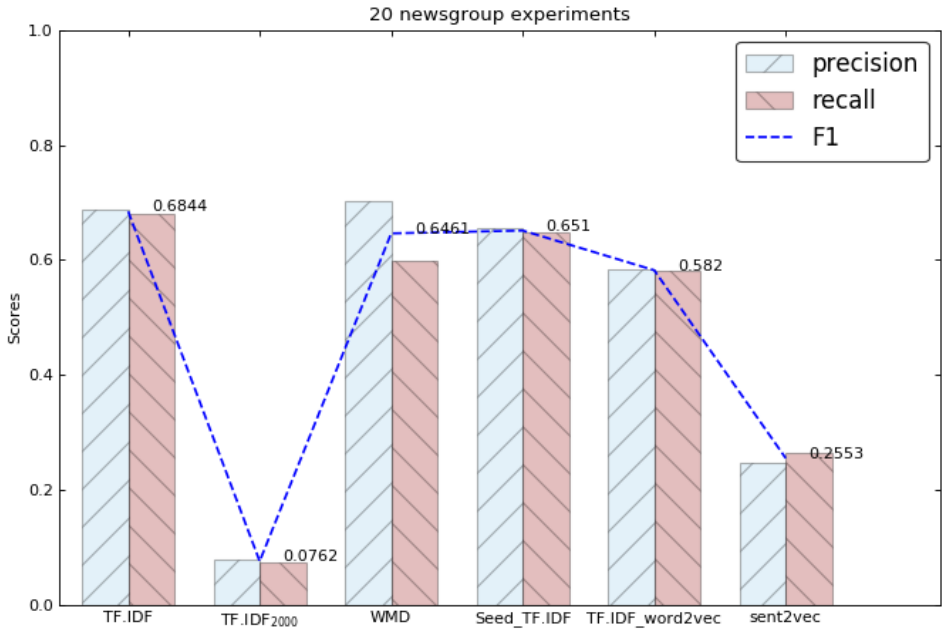

Figure 8.2: The results of five methods in 20 newsgroup experiments

8.2.1 Datasets

Ren_CECps In this paper, we used all of the annotated sentences for experiments. The total number of the sentences is 36163, and we selected 6163 sentences randomly as WMD seed corpus, the rest part of the corpus was divided into training data and test data each has 15000 sentences selected randomly.

WMD Vector For the sentences in training and test data, using WMD algorithm to calculate the distances among target sentence and selected seed corpus. That means the feature vector in this paper is 6163 and the NaN results are all converted into 0.

Decision Tree We utilized a "DecisionTreeClassifier" package in sklearn tools for baseline experiments[8].

8.2.2 Results

The deep neural network output the probabilities of nine emotion labels. To get final category labels, we run a threshold selection algorithm based on maximizing the f1 scores of every single emotional categories. The selected thresholds for nine emotion categories are

[8]http://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html

0.073, 0.261, 0.121, 0.099, 0.187, 0.366, 0.084, 0.261, 0.033 corresponding to the emotions of 'anger', 'anxiety', 'expect', 'hate', 'joy', 'love', 'neutral', 'sorrow', 'surprise' respectively. According to the those thresholds, we can get the classification results in Table 8.2.

Table 8.2: The classification results of deep neural network and decision tree

Algorithm	**Case 1**	**Case 2**	**Case 3**
		precision	
Deep neural network	1.0	0.746	0.022
Decision tree	0.994	0.369	0.121

Note: Case 1 means recognizing emotion label or unrecognizing emotion label; Case 2 means recognizing at less one emotion label(single match); Case 3 means recognizing all of the truth emotion labels(full match).

In Table 8.2, we can find that both deep neural network and decision tree have the capacity of recognizing whether a sentence has emotion or not, and deep neural network get higher result. When comes to single match experiments, the advantage of deep neural network is obviously demonstrated and gets twice times higher precision than decision tree. But in Case 3 of full match situation, decision tree get higher precision than deep neural network.

8.3 Evaluation of Emotional Trigger System

8.3.1 Setup

In this part, we will very the classification abilities among fast WMD, CNN+LSTM based network, CNN based network and LSTM based network using the split 1v1 and 4v1 data sets of Ren_CECps. The time consuming experiments among these algorithms follow the same ten times' computing rule. We randomly select 30 sentences from Ren_CECps, manually making the first 10, second 10 and the last 10 sentences as the ten times' computing experiment data respectively. The seed corpus needed for fast WMD keeps the same with previous experiments.

To make a sententious declare for the experiments above, we give the abbreviations as follows:

- **WMD_1v1:** the fast WMD method based on 1v1 data set;
- **WMD_4v1:** the fast WMD method based on 4v1 data set;
- **cnn_lstm_1v1:** the CNN+LSTM network based on 1v1 data set;
- **cnn_lstm_4v1:** the CNN+LSTM network based on 4v1 data set;
- **cnn_1v1:** the CNN network based on 1v1 data set;
- **cnn_4v1:** the CNN network based on 4v1 data set;
- **lstm_1v1:** the LSTM network based on 1v1 data set;
- **lstm_4v1:** the LSTM network based on 4v1 data set;

8.3.2 Hyperparameters

In the experiments, we use keras[16] as the construction API for the DNNS and the backend is TensorFlow[1]. All of the sentences are preprocessed into sequences with the words presented by their indexes in lexicon. The vocabulary length is 32126. Table 8.3 shows the lengths of sentences in Ren_CECps. The statistics shows more than 99.9% of the sentences have the length less than 200, thus the "maxlen" of padding is selected as 200.

Table 8.3: The lengths of sentences in Ren_CECps

length	**total**	**0-200**	**200-300**	**300-500**
sentence No.	50321	50312	7	2
per. (%)	100	99.982	0.0139	0.0041

Limited by the scale of the corpus, embedding size is set to 128, 0.5 for the dropout percentage. For the convolution layer, kernel size is 5, calculated with 64 filters. The corresponding MaxPooling layer gets 4 as the pool size. We make the output size of LSTM layer as 70, and for batch size, 30 is selected. All of the parameters are chosen empirically.

Other parameters for Conv1D layer are "padding" with "valid", "activation" with "relu" and "strides" with "1". For the last dense layer, we make it a nine cells with

"RandomUniform" as "kernel_initializer" corresponding to the loss parameter of "categorical_crossentropy" on nine emotion categories. The activation function for dense layer is "softmax". Using "adam" to optimize loss function and "accuracy" for metrics of the network. The epochs is 3 exported through experiments, this will be explained in the discussion section.

8.3.3 Results

Relying on the parameters and corpus, we can get the time consuming results of fast WMD and three networks in Table 8.4. Fig.8.3 shows the acceleration lines with log scaled on the values. Table 8.5 gives the precisions, recalls and F1-scores of fast WMD and three networks over the "micro" and "macro" standards, the Fig.8.4 is the tendency graph based on these data.

Table 8.4: The time-consuming experiments among fast WMD and three networks

Groups	**Case 1**	**Case 2**	**Case 3**	**Average**
	per 10 times(s)			
WMD_1v1	**66.26106**	84.76050	**77.94079**	76.3207
WMD_4v1	66.26100	**86.76047**	77.94075	76.9874
cnn_lstm_1v1	0.01401	0.01770	0.01348	0.0150
cnn_1v1	**0.00557**	**0.00495**	**0.00565**	0.0053
lstm_1v1	0.02908	0.02992	0.03078	0.0299
cnn_lstm_4v1	0.01458	0.01292	0.01167	0.0130
cnn_4v1	0.00866	0.00975	0.00588	0.0080
lstm_4v1	0.02605	0.02784	0.02817	0.0273

In Table 8.4, we can clearly find it's a thousandfold acceleration after using deep neural networks. For an average time in one ten times' computing, the fast WMD needs 76s, that's 7 second per recognition the same with our previous experiments. LSTM based model needs the most times compared with CNN and CNN+LSTM based models, it's 0.03s totally and 3 millisecond per recognition in average. In Fig.8.3, we draw the slate blue guide line as the lower limit based on the LSTM model, and the upper limit based on fast WMD. We get 2278 times, 2832 times and 2531 times during three ten times' computing. The great promotion in the times gives our emotional trigger system a real

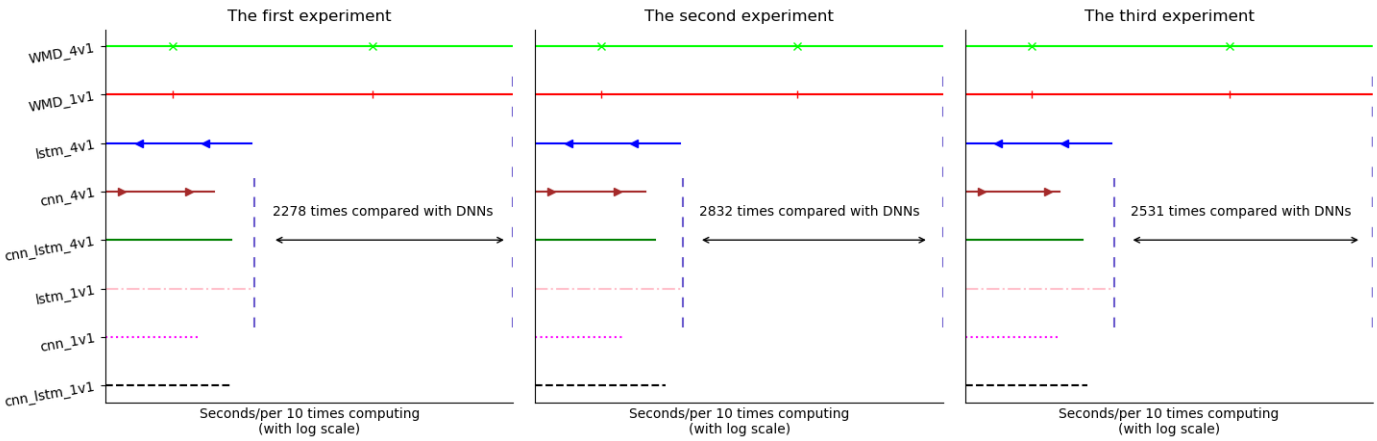

Figure 8.3: The acceleration results among fast WMD and three networks

time response ability. In this respect, the CNN, LSTM and CNN+LSTM based networks are high-efficiency to solve the time delay problem.

On the other hand, though the DNNs get the fastest emotional trigger, without higher or at least similar classification ability, the speed is useless. Table 8.5 and Fig.8.4 show the verification experiments for this doubt. In Fig.8.4, we draw two histograms to represent precision and recall separately, in which red color with slash texture means precision, tan color with vertical texture denotes recall and F1 scores are drafted as polyline with lime color. The results of 1v1 and 4v1 are placed together by 4 sub-figures, the first row represents 1v1 experiments and second row represents 4v1 experiments.

In the experiments, we measure the results based on Micro and Macro standard respectively. Considering the Micro side, in 1v1 experiments, the CNN+LSTM gets the best f1 score compared with other methods of 0.35002, all of the three networks get the f1 scores over 30% and overing 12 percentages than fast WMD model. When comes with more training data in 4v1 experiments, fast WMD gets the best f1 score, 13 percentages over its 1v1 results and 0.1 percentage higher than the best f1 score achieved by CNN+LSTM model. With more training data, the fast WMD model gets obvious promotion, while the CNN and LSTM models keep the same, only CNN+LSTM network moves forward.

But in Macro standard, the results go to the opposite side. In Fig.8.4, we can see the fast WMD gets the best f1 scores both in 1v1 and 4v1 experiments. In 1v1 experiment, the three networks get almost the same f1 scores 7 percentages lower than the fast WMD.

Table 8.5: The classification results on fast WMD and three networks

Type	Algorithm	Precision	Recall	F1-score
Micro	WMD_1v1	0.23887	0.23887	0.23887
	cnn_lstm_1v1	0.35002	0.35002	**0.35002**
	cnn_1v1	0.30734	0.30734	0.30734
	lstm_1v1	0.33647	0.33647	0.33647
	WMD_4v1	0.36759	0.36759	**0.36759**
	cnn_lstm_4v1	0.36600	0.36600	**0.36600**
	cnn_4v1	0.30596	0.30596	0.30596
	lstm_4v1	0.33618	0.33618	0.33618
Macro	WMD_1v1	0.35858	0.27378	**0.31050**
	cnn_lstm_1v1	0.23938	0.23752	0.23844
	cnn_1v1	0.23065	0.23413	0.23237
	lstm_1v1	0.25158	0.23760	0.24439
	WMD_4v1	0.33847	0.30025	**0.31822**
	cnn_lstm_4v1	**0.38705**	0.23755	**0.29441**
	cnn_4v1	0.23624	0.23696	0.23660
	lstm_4v1	0.26606	0.24427	0.25470

In 4v1 experiments, though fast WMD still get the best f1 score, its promotion only goes 0.8 percentage. As the comparison, CNN model gets the least 0.4% improvement, LSTM model gets the promotion of one percentage and CNN+LSTM model achieves the most improvement of 7%. In this experiment, CNN+LSTM network gets the best precision of 0.387 among all of the algorithms.

Considering the Micro and Macro results, we can find that Both in these two calculation standard, f1 scores of CNN+LSTM method are improved from 1v1 to 4v1 obviously. The fast WMD only changes clearly in Micro experiments, LSTM netowrk makes a small step only in Macro. CNN based network almost keep the same results from 1v1 to 4v1 either in Micro or Macro.

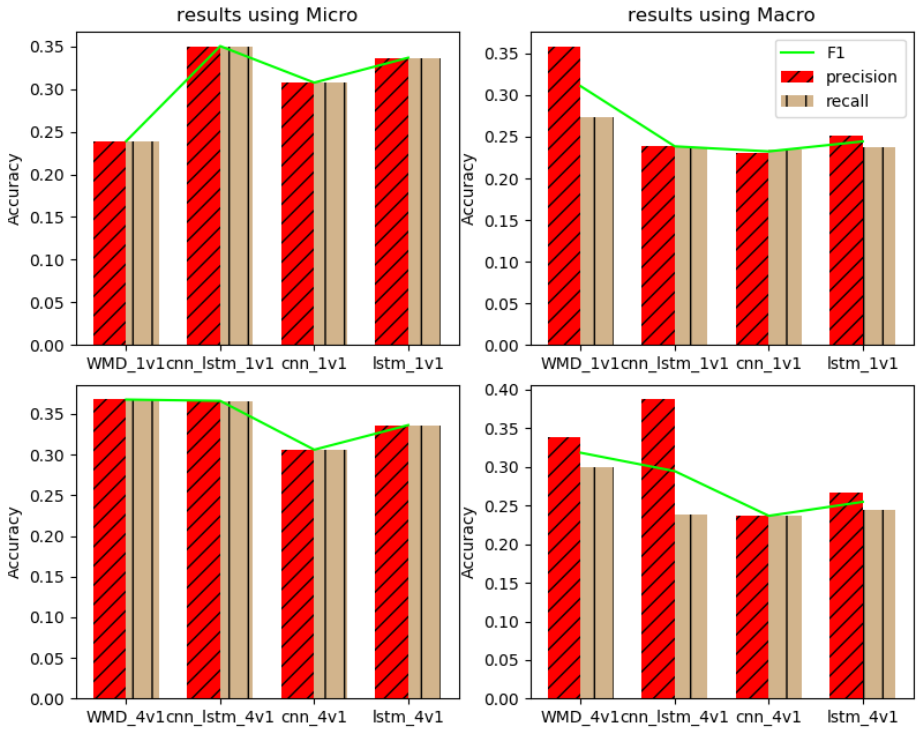

Figure 8.4: The classification results on fast WMD and three networks

In the results of time consuming experiments and classification comparisons, the CNN+LSTM network prove its ability to make a real time emotional trigger with acceptable accuracy compared with fast WMD. Thus, the Actroid REN-XIN can make an emotional action with ignorable time delay over the whole recognition and response processes when handling with several scenes.

Chapter 9

Conclusions

9.1 Discussion in WMD Based Models

Difficulty in SeTF · IDF Though SeTF · IDF can match sentences into different emotion dimensions, the method is based on priori knowledges annotated in corpus. This means we cannot use SeTF · IDF to match a new sentence or document into multi-emotion dimensions due to lack of no emotional keywords annotated manually. That's why we use SeTF · IDF as an enhanced baseline method. It's an idealized results. The importance is this visualization algorithm makes us having a clearer visual results, and changes our way of thinking in training multi-label data.

"NaN" conversion Both of the results of the Chinese and English corpus computed by WMD contain "NaN" data. This makes the data disable to train in SVM model. Parsing the sentence pairs which "NaN" data happened, we found the "NaN" data always appear in short sentences or documents, and the target data are the same with the seed data, like " (English: good)" and " (English: I don't know.)" in Chinese corpus, "Thanks!" and "It's there..." in English corpus. One special situation is the documents with a lot of messy script codes in 20 newsgroup data set, and these messy codes will result in "NaN" data.

Having known the contents leaded to error, we make two ways to convert the "NaN" data. One is replacing the "NaN" data with "0", and the other one is "1". The reason for "0" conversion is that the pairs of sentences are the same, and in distance meanings of

WMD, ”0” is the most suitable and practical; But considering the similarity vector, ”0” elements are useless, and may cause information loss, ”1” conversion maybe better.

To verify which one is suitable, we convert the ”NaN” data to ”0” and ”1” independently in both 1v1 and 4v1 experiments. The ”1” conversion data gets 0.291 and 0.305 in F1-scores in 1v1 and 4v1 experiments respectively, a bit lower than ”0” conversion of 0.310 and 0.318. Thus, we choose ”0” conversion finally in all of the experiments.

The opposite results in Chinese and English data sets In 1v1 and 4v1 experiments, the WMD method gets the best results. On the contrary, in 20 news, the TF · IDF gets the best result, and Seed_TF · IDF gets the second, third is WMD. One reason is the reduction rates mentioned above of the two language corpus are different. The other reason maybe the word embeddings of English used in the experiments have more missing words than Chinese embeddings, this can explain why TF · IDF_word2vec gets higher results and indeed should be higher than TF · IDF in Chinese corpus, but gets fourth rank in English corpus and almost 10% lower than TF · IDF.

9.2 Discussion in Emotional Trigger System

The proposed CNN+LSTM network successfully solved the long time delay with trainable accuracy, there still are some key points needed to be discussed to give an experience for others or related field research works.

epochs When dealing with parameter of epochs, we run a 50 epochs experiment in 1v1 and 4v1 data sets among all the three networks. The tendency of accuracy and loss changed with epochs can be seen in Fig 8. Red line denotes the accuracy of training data, tan dot line means the loss calculated though training, lime line represents the testing accuracy and blue dot line is the corresponding loss.

In Fig.9.1, we annotated the guide lines of $epochs = 3$ in every sub-figures. And in model training process, the epochs of the three networks are all set to 3 according to this guide line, this is not an empirical selection. We can check the accuracy and loss pairs in every sub-figures, the inflection point can be obviously found in $epochs = 2$. All of the six sub-figures show the increase of training accuracies with the decrease of training losses

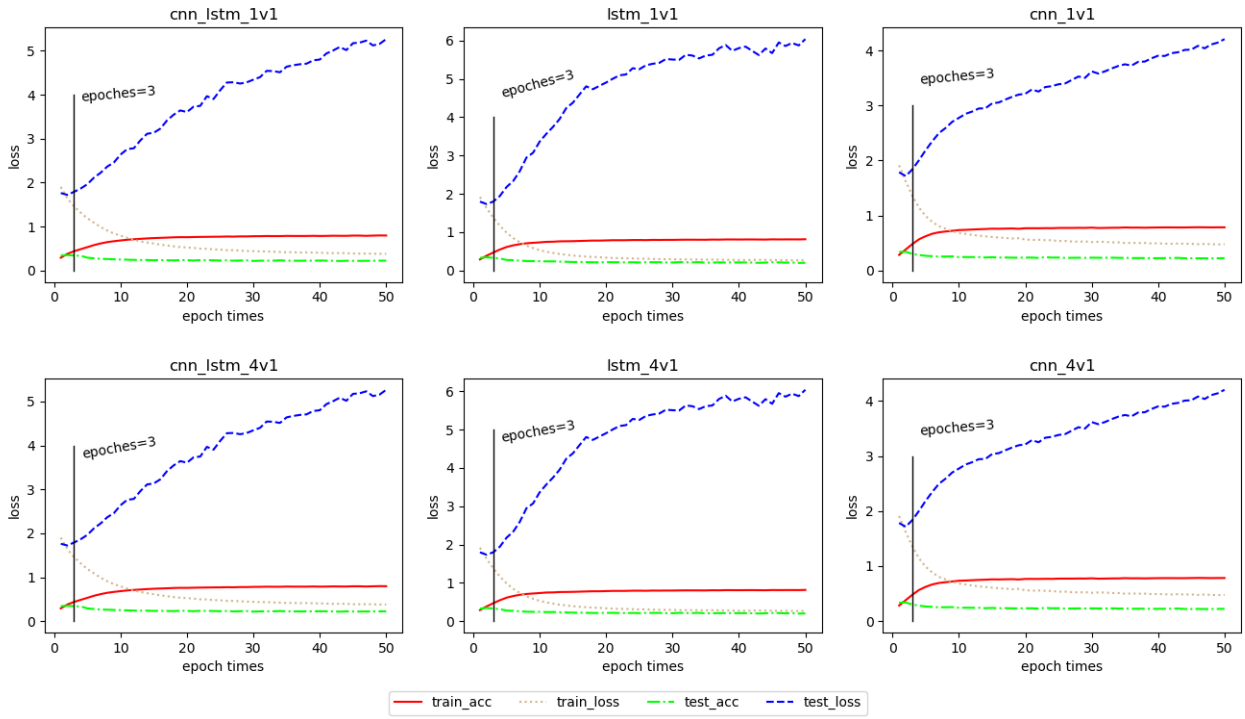

Figure 9.1: The tendency of accuracy and loss changed with epochs in three networks

epoch by epoch, while the test accuracies keep declining with test loss keeping raising till no changes unfortunately.

In experiences, the *epochs* = 2 should be the suitable parameter for training. With the contrary tendencies obtained from the six graphs, we make a speculation that the former experiences maybe not fit this multi-class emotional corpus well. To very this hypothesis, we save the models after every epochs during our 50 times iteration, and then calculate the f1 scores in Macro standard over every exported epoch based models (the metrics parameters in keras are all measured by Micro standard). The results show that *epochs* = 2 gets the best precision in Micro standard which can be seen clearly in Fig 8, while *epochs* = 3 gets the best f1 scores. In Micro standard, *epochs* = 3 gets both the best precision and f1 score than other epochs. According to these comparative trials, we finally make the *epochs* as 3.

Different promotion among four methods In Table 8.5 and Fig.8.4, we can find the fast WMD performs well in Macro standard, CNN+LSTM method achieves better in Micro. The results among three networks also show different tendencies. These are

interesting.

We dig from the start: the data split in Ren_CECps. The sentences are all annotated with 8 emotional labels mentioned before. With no emotional labels, the intensity values of corresponding categories will be 0, and we written the category of these sentences as "neutral". In the process of splitting data into single category level, we add the sentence to the specific emotion if its intensity value is not 0, that means one sentence can appear in more than one categories sub-data sets, one the other side, the "neutral" sentences can only appear in one sub-data set.

In Ren_CECps, there are 36525 sentences, 22751 sentences have only one emotional category, 62.28% of the total, 11731 sentences have two emotional labels, 32.11% of the total and 1847 sentences have three labels, 5.05% of the total. The 1v1 and 4v1 data sets are randomly selected based on the sub-data sets, the selected training and testing data in 1v1 and 4v1 are all have the same sentences with different emotional labels according to the 37.16% of sentences are two or three labels annotated. Thinking in this way, we can call our experiments the multi-class classification with adversarial samples.

We firstly analyze the fast WMD method with CNN+LSTM network. The contrary tendencies in Micro and Macro results show the different feature representation abilities. In the adversarial data set, the fast WMD calculates the features from words to words with distance measurement which is better than sparse cells learning in CNN+LSTM network, and results to 0.31 vs 0.23 in 1v1 and 0.31 vs 0.29 in 4v1 under Macro standard. While under Micro, the CNN+LSTM model shows the stronger ability to learn global features than fast WMD, this results to the 10 percentages in 1v1 experiments. With more data, the fast WMD achieves 0.1% higher score than CNN+LSTM and the f1 scores changes from 0.23 to 0.36, shows fast WMD can learn more global features than CNN+LSTM against with more adversarial samples though the training data are also increased. The inconspicuous improvement of CNN+LSTM shows that the network is sensitive with adversarial samples and can not learn more global features. Back to the Macro, the stronger ability of feature representation of fast WMD can not achieve more promotion either, while CNN+LSTM gets 6 percentages increase. Combining the Micro and Macro results, we can find that with the increase of training data, CNN+LSTM network can not learn global features, but the ability of learning local feature of categories are still working and

achieves obviously promotion. With more training data, the CNN++LSTM network can learn both global and local feature better than fast WMD method.

Next we analyze the three networks. The almost unaltered scores in Micro shows the three networks are all sensitive and can not learn more global features from more data with adversarial samples. In Macro, the separated experiments employed each networks show what the local feature learning ability rely on. We can find in Table 8.5, the CNN based network achieves almost the same scores of 0.232 to 0.236, the LSTM based network achieves 1 percentages improvement of 0.244 to 0.254. The different promotions achieved between CNN and LSTM networks can be a demonstration that LSTM layer has a strong ability to learn local features of categories against adversarial samples than CNN, that's the ability of strong sequence information learning. But the results will not give the death of CNN, in the CNN+LSTM, the 6 percentages achievement and also the best results among three networks show the CNN can filter information of adversarial samples and enhance the learning ability of LSTM, thus results to the better learning ability of local features than fast WMD.

9.3 Conclusion and Future Work

The experiments of WMD algorithm show some enlightening conclusions based on the cross-language corpus.

1. Distance changed by the emotion separated method can get higher visual performance in multi-label emotional corpus;

2. The WMD algorithm is indeed efficient for classification.

3. Different language has different information density of words. Thus can influence the results of feature representation methods.

4. In Chinese corpus, owing to the high information density of words, for a certain degree of feature representation reduction, it's good for the classifier to training models and can help improve results; For English corpus, due to the lower information density of words, the same degree for reducing features may not good for model training and needs more experiments to find the best degree.

According the comparison experiments and discussions of emotional trigger system, we can make the conclusion below:

1. We proposed a fast WMD based emotional trigger system which can extend the Humanoid robot interaction for more emotion control actions;

2. The CNN+LSTM can obviously speed up the emotional trigger system without emotion classification accuracy descent;

3. CNN+LSTM based network has a stronger ability to train models against adversarial samples than CNN only or LSTM only based networks;

4. For small corpus training using deep neural networks, the inflection point of training and testing is not always the best parameter, the next epoch maybe better.

The CNN+LSTM model show the strong ability to learn global feature for multi-class corpus, though the local features are not performed as expected . And in Section 8.2.2, the case 1 and case 2 experiments show the deep neural network has strong ability to recognize emotions, but in case 3, the decision tree get higher precision. These all give us a conclusion that simple deep neural network is good at simple scenario, bur for complex emotion expression, there still need more research.

For the emotion classification experiments, the just over 30% F1-scores cannot supply the emotional recognition applications, we will continue to improve this field. In the future, we will focus to train the networks for single emotional categories separately without adversarial samples. Some attention mechanism based networks will be proposed to train the emotion keywords in annotated corpus for finding the emotion action points in contents. For the Actroid REN-XIN, we need more research to make the robot a better auto-response ability without manual operated controls of emotional expression.

Bibliography

[1] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[2] A. Andreevskaia and S. Bergler. Mining wordnet for a fuzzy sentiment: Sentiment tag extraction from wordnet glosses. In *EACL*, volume 6, pages 209–216, 2006.

[3] G. Andrew. A hybrid markov/semi-markov conditional random field for sequence segmentation. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 465–472. Association for Computational Linguistics, 2006.

[4] A. Arleo, W. Didimo, G. Liotta, and F. Montecchiani. Large graph visualizations using a distributed computing platform. *Information Sciences*, 381:124–141, 2017.

[5] S. Baccianella, A. Esuli, and F. Sebastiani. Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *LREC*, volume 10, pages 2200–2204, 2010.

[6] M. S. Bartlett, G. Littlewort, I. Fasel, and J. R. Movellan. Real time face detection and facial expression recognition: Development and applications to human computer interaction. In *Computer Vision and Pattern Recognition Workshop, 2003. CVPRW'03.*, volume 5, pages 53–53. IEEE, 2003.

[7] K. Berns and J. Hirth. Control of facial expressions of the humanoid robot head roman. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3119–3124. IEEE, 2006.

[8] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.

[9] A. Buja, D. F. Swayne, M. L. Littman, N. Dean, H. Hofmann, and L. Chen. Data visualization with multidimensional scaling. *Journal of Computational and Graphical Statistics*, 17(2):444–472, 2008.

[10] A. M. d. J. C. Cachopo. Improving methods for single-label text categorization. *Instituto Superior Técnico, Portugal*, 2007.

[11] E. Cambria. Affective computing and sentiment analysis. *IEEE Intelligent Systems*, 31(2):102–107, 2016.

[12] E. Cambria, D. Olsher, and D. Rajagopal. Senticnet 3: a common and common-sense knowledge base for cognition-driven sentiment analysis. In *Twenty-eighth AAAI conference on artificial intelligence*, 2014.

[13] N. Cao, Y.-R. Lin, and D. Gotz. Untangle map: Visual analysis of probabilistic multi-label data. *IEEE transactions on visualization and computer graphics*, 22(2):1149–1163, 2016.

[14] X. Cheng, Y. Chen, B. Cheng, S. Li, and G. Zhou. An emotion cause corpus for chinese microblogs with multiple-user structures. *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)*, 17(1):6, 2017.

[15] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[16] F. Chollet et al. Keras. `https://github.com/fchollet/keras`, 2015.

[17] G. G. Chowdhury. Natural language processing. *Annual review of information science and technology*, 37(1):51–89, 2003.

[18] C. Cortes and V. Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[19] E. S. Dan-Glauser and K. R. Scherer. The difficulties in emotion regulation scale (ders): Factor structure and consistency of a french translation. *Swiss Journal of Psychology*, 72(1):5, 2013.

[20] M. De Choudhury. Anorexia on tumblr: A characterization study. In *Proceedings of the 5th International Conference on Digital Health 2015*, pages 43–50. ACM, 2015.

[21] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition, 2009. CVPR 2009*, pages 248–255. Ieee, 2009.

[22] Y. Fan, X. Lu, D. Li, and Y. Liu. Video-based emotion recognition using cnn-rnn and c3d hybrid networks. In *Proceedings of the 18th ACM International Conference on Multimodal Interaction*, pages 445–450. ACM, 2016.

[23] D. R. Faria, M. Vieira, F. C. Faria, and C. Premebida. Affective facial expressions recognition for human-robot interaction. In *IEEE RO-MAN'17: IEEE International Symposium on Robot and Human Interactive Communication, Lisbon, Portugal*, 2017.

[24] J. R. Finkel and C. D. Manning. Joint parsing and named entity recognition. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 326–334. Association for Computational Linguistics, 2009.

[25] M. Friendly and D. Denis. The early origins and development of the scatterplot. *Journal of the History of the Behavioral Sciences*, 41(2):103–130, 2005.

[26] A. Y. Fu, L. Wenyin, and X. Deng. Detecting phishing web pages with visual similarity assessment based on earth mover's distance (emd). *IEEE transactions on dependable and secure computing*, 3(4), 2006.

[27] L. Giles et al. *Taoist teachings*, volume 5, pages 90–91. Library of Alexandria, 1912.

[28] X. Glorot, A. Bordes, and Y. Bengio. Domain adaptation for large-scale sentiment classification: A deep learning approach. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 513–520, 2011.

[29] R. E. Goldsmith and D. Horowitz. Measuring motivations for online opinion seeking. *Journal of interactive advertising*, 6(2):2–14, 2006.

[30] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[31] T. W. Group et al. Guidelines for temporal expression annotation for english for tempeval 2010, 2009.

[32] L. Gui, J. Hu, Y. He, R. Xu, Q. Lu, and J. Du. A question answering approach to emotion cause extraction. *arXiv preprint arXiv:1708.05482*, 2017.

[33] L. Gui, L. Yuan, R. Xu, B. Liu, Q. Lu, and Y. Zhou. Emotion cause detection with linguistic construction in chinese weibo text. In *Natural Language Processing and Chinese Computing*, pages 457–464. Springer, 2014.

[34] T. Hashimoto, S. Hitramatsu, T. Tsuji, and H. Kobayashi. Development of the face robot saya for rich facial expressions. In *SICE-ICASE, 2006. International Joint Conference*, pages 5423–5428. IEEE, 2006.

[35] F. Heimerl, S. Koch, H. Bosch, and T. Ertl. Visual classifier training for text document retrieval. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2839–2848, 2012.

[36] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[37] M. Hu and B. Liu. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 168–177. ACM, 2004.

[38] X. Huang, W. Lai, A. Sajeev, and J. Gao. A new algorithm for removing node overlapping in graph visualization. *Information Sciences*, 177(14):2821–2844, 2007.

[39] A. Inselberg. The plane with parallel coordinates. *The visual computer*, 1(2):69–91, 1985.

[40] L. Jian, L. Yang, and W. Suge. The constitution of a fine-grained opinion annotated corpus on weibo. In *Chinese Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data*, pages 227–240. Springer, 2016.

[41] T. Joachims. A probabilistic analysis of the rocchio algorithm with tfidf for text categorization. Technical report, Carnegie-mellon univ pittsburgh pa dept of computer science, 1996.

[42] H. Joho, A. Jatowt, and R. Blanco. Ntcir temporalia: a test collection for temporal information access research. In *Proceedings of the companion publication of the 23rd international conference on World wide web companion*, pages 845–850. International World Wide Web Conferences Steering Committee, 2014.

[43] H. Joho, A. Jatowt, R. Blanco, H. Yu, and S. Yamamoto. Overview of NTCIR-12 temporal information access (temporalia-2) task. In *Proceedings of the 12th NTCIR Conference on Evaluation of Information Access Technologies, June 7-10, 2016, Tokyo, Japan*, 2016.

[44] H. Joho, A. Jatowt, and B. Roi. A survey of temporal web search experience. In *Proceedings of the 22nd international conference on World Wide Web companion*, pages 1101–1108. International World Wide Web Conferences Steering Committee, 2013.

[45] X. Ke, Y. Yang, and J. Xin. Facial expression on robot shfr-iii based on head-neck coordination. In *2015 IEEE International Conference on Information and Automation*, pages 1622–1627. IEEE, 2015.

[46] Y. Kim. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*, 2014.

[47] Y. Kim, Y. Jernite, D. Sontag, and A. M. Rush. Character-aware neural language models. In *AAAI*, pages 2741–2749, 2016.

[48] Y. Kim, H. Lee, and E. M. Provost. Deep learning for robust feature generation in audiovisual emotion recognition. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3687–3691. IEEE, 2013.

[49] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[50] M. Kusner, Y. Sun, N. Kolkin, and K. Weinberger. From word embeddings to document distances. In *International Conference on Machine Learning*, pages 957–966, 2015.

[51] Q. Le and T. Mikolov. Distributed representations of sentences and documents. In *International Conference on Machine Learning*, pages 1188–1196, 2014.

[52] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989.

[53] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[54] J. Leskovec, A. Rajaraman, and J. D. Ullman. *Mining of massive datasets*. Cambridge university press, 2014.

[55] J. Li and F. Ren. Emotion recognition of weblog sentences based on an ensemble algorithm of multi-label classification and word emotions. *IEEJ Transactions on Electronics, Information and Systems*, 132(8):1362–1375, 2012.

[56] T. M. Li, M. Chau, P. S. Yip, and P. W. Wong. Temporal and computerized psycholinguistic analysis of the blog of a chinese adolescent suicide. *Crisis*, 2014.

[57] B. Liu. Sentiment analysis and opinion mining. *Synthesis lectures on human language technologies*, 5(1):1–167, 2012.

[58] N. Liu, M. He, C. Li, X. Kang, and F. Ren. Tuta1 at the ntcir-12 temporalia task. In *NTCIR*, 2016.

[59] D. E. Losada, F. Crestani, and J. Parapar. Clef 2017 erisk overview: Early risk prediction on the internet: Experimental foundations. In *CEUR Workshop Proceedings*, 2017.

[60] Y. Lu, K. Sakamoto, H. Shibuki, and T. Mori. Construction of a multilingual annotated corpus for deeper sentiment understanding in social media. *Journal of Natural Language Processing*, 24(2):205–265, 2017.

[61] Y. Ma, J. Xu, X. Wu, F. Wang, and W. Chen. A visual analytical approach for transfer learning in classification. *Information Sciences*, 390:54–69, 2017.

[62] L. v. d. Maaten and G. Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(Nov):2579–2605, 2008.

[63] R. Mahum, F. Shafique Butt, K. Ayyub, S. Islam, M. Nawaz, and D. Abdullah. A review on humanoid robots. In *International Journal of ADVANCED AND APPLIED SCIENCES*, volume 4, pages 83–90, 02 2017.

[64] K. Matsumoto, K. Kita, and F. Ren. Emotion estimation of wakamono kotoba based on distance of word emotional vector. In *7th International Conference on Natural Language Processing and Knowledge Engineering (NLP-KE)*, pages 214–220. IEEE, 2011.

[65] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[66] T. Mikolov, M. Karafiát, L. Burget, J. Černockỳ, and S. Khudanpur. Recurrent neural network based language model. In *Eleventh Annual Conference of the International Speech Communication Association*, 2010.

[67] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[68] J. Minato, D. Bracewell, F. Ren, and S. Kuroiwa. Statistical analysis of a japanese emotion corpus for natural language processing. *Computational Intelligence*, pages 924–929, 2006.

[69] M. Minsky. *Society of mind*. Simon and Schuster, 1988.

[70] S. M. Mohammad. #Emotional tweets. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics-Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation*, pages 246–255. Association for Computational Linguistics, 2012.

[71] S. M. Mohammad, F. Bravo-Marquez, M. Salameh, and S. Kiritchenko. Semeval-2018 Task 1: Affect in tweets. In *Proceedings of International Workshop on Semantic Evaluation (SemEval-2018)*, New Orleans, LA, USA, 2018.

[72] T. K. Moon. The expectation-maximization algorithm. *IEEE Signal processing magazine*, 13(6):47–60, 1996.

[73] S. Mukherjee and P. Bhattacharyya. Feature specific sentiment analysis for product reviews. *Computational Linguistics and Intelligent Text Processing*, pages 475–487, 2012.

[74] S. A. Najim and I. S. Lim. Trustworthy dimension reduction for visualization different data sets. *Information Sciences*, 278:206–220, 2014.

[75] J. Needham. *Science and Civilization in China*, volume 2, page 53. England: Cambridge University Press, 1986.

[76] J. Y.-H. Ng, J. Choi, J. Neumann, and L. S. Davis. Actionflownet: Learning motion representation for action recognition. *arXiv preprint arXiv:1612.03052*, 2016.

[77] L. Nurul, S. Sakriani, Y. Koichiro, and N. Satoshi. Eliciting positive emotion through affect-sensitive dialogue response generation: A neural network approach. In *The Thirty-Second AAAI Conference on Artificial Intelligence (AAAI-18)*, pages 5293–5300, 2018.

[78] M. Pagliardini, P. Gupta, and M. Jaggi. Unsupervised Learning of Sentence Embeddings using Compositional n-Gram Features. *arXiv*, 2017.

[79] B. Pang and L. Lee. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the 43rd annual meeting on association for computational linguistics*, pages 115–124. Association for Computational Linguistics, 2005.

[80] V. Patel, A. J. Flisher, S. Hetrick, and P. McGorry. Mental health of young people: a global public-health challenge. *The Lancet*, 369(9569):1302–1313, 2007.

[81] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[82] O. Pele and M. Werman. Fast and robust earth mover's distances. In *IEEE 12th international conference on Computer vision*, pages 460–467. IEEE, 2009.

[83] J. Pennington, R. Socher, and C. Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[84] A. Petropoulos, S. P. Chatzis, and S. Xanthopoulos. A hidden markov model with dependence jumps for predictive modeling of multidimensional time-series. *Information Sciences*, 2017.

[85] R. W. Picard and R. Picard. *Affective computing*, volume 252. MIT press Cambridge, 1997.

[86] F. J. Pineda. Generalization of back-propagation to recurrent neural networks. *Physical review letters*, 59(19):2229, 1987.

[87] C. Quan and F. Ren. A blog emotion corpus for emotional expression analysis in chinese. *Computer Speech & Language*, 24(4):726–749, 2010.

[88] C. Quan and F. Ren. Sentence emotion analysis and recognition based on emotion words using ren-cecps. *International Journal of Advanced Intelligence*, 2(1):105–117, 2010.

[89] C. Quan and F. Ren. Weighted high-order hidden markov models for compound emotions recognition in text. *Information Sciences*, 329:581–596, 2016.

[90] R. Řehůřek and P. Sojka. Software framework for topic modelling with large corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta, May 2010. ELRA.

[91] F. Ren and Z. Huang. Automatic facial expression learning method based on humanoid robot xin-ren. *IEEE Transactions on Human-Machine Systems*, 46(6):810–821, 2016.

[92] F. Ren and X. Kang. Employing hierarchical bayesian networks in simple and complex emotion topic analysis. *Computer Speech & Language*, 27(4):943–968, 2013.

[93] F. Ren, X. Kang, and C. Quan. Examining accumulated emotional traits in suicide blogs with an emotion topic model. *IEEE journal of biomedical and health informatics*, 20(5):1384–1396, 2016.

[94] F. Ren and N. Liu. Emotion computing using word mover's distance features based on ren_cecps. *PloS one*, 13(4):1–17, 2018.

[95] I. Rish et al. An empirical study of the naive bayes classifier. In *IJCAI 2001 workshop on empirical methods in artificial intelligence*, volume 3(22), pages 41–46. IBM New York, 2001.

[96] Y. Rubner, C. Tomasi, and L. J. Guibas. A metric for distributions with applications to image databases. In *Sixth International Conference on Computer Vision*, pages 59–66. IEEE, 1998.

[97] Y. Rubner, C. Tomasi, and L. J. Guibas. The earth mover's distance as a metric for image retrieval. *International journal of computer vision*, 40(2):99–121, 2000.

[98] D. E. Rumelhart, J. L. McClelland, P. R. Group, et al. *Parallel distributed processing*, volume 1. MIT press Cambridge, MA, 1987.

[99] G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. *Information processing & management*, 24(5):513–523, 1988.

[100] G. Salton, A. Wong, and C.-S. Yang. A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620, 1975.

[101] N. Sebe, M. S. Lew, I. Cohen, A. Garg, and T. S. Huang. Emotion recognition using a cauchy naive bayes classifier. In *2002. Proceedings. 16th International Conference on Pattern Recognition*, volume 1, pages 17–20. IEEE, 2002.

[102] D. Spiegelhalter, M. Pearson, and I. Short. Visualizing uncertainty about the future. *science*, 333(6048):1393–1400, 2011.

[103] G. W. Stewart. On the early history of the singular value decomposition. *SIAM review*, 35(4):551–566, 1993.

[104] Q. Su, X. Xu, H. Guo, Z. Guo, X. Wu, X. Zhang, B. Swen, and Z. Su. Hidden sentiment association in chinese web opinion mining. In *Proceedings of the 17th international conference on World Wide Web*, pages 959–968. ACM, 2008.

[105] S. Tan and J. Zhang. An empirical study of sentiment analysis for chinese documents. *Expert Systems with applications*, 34(4):2622–2629, 2008.

[106] D. Tang, F. Wei, B. Qin, N. Yang, T. Liu, and M. Zhou. Sentiment embeddings with applications to sentiment analysis. *IEEE Transactions on Knowledge and Data Engineering*, 28(2):496–509, 2016.

[107] X. Wan. A novel document similarity measure based on earth mover's distance. *Information Sciences*, 177(18):3718–3730, 2007.

[108] C. Wang, C. Quan, and F. Ren. Maximum entropy based emotion classification of chinese blog sentences. In *International Conference on Natural Language Processing and Knowledge Engineering (NLP-KE)*, pages 1–7. IEEE, 2010.

[109] C. Wang, M. Zhang, S. Ma, and L. Ru. Automatic online news issue construction in web environment. In *Proceedings of the 17th international conference on World Wide Web*, pages 457–466. ACM, 2008.

[110] L. Wang, F. Ren, and D. Miao. Multi-label emotion recognition of weblog sentence based on bayesian networks. *IEEJ Transactions on Electrical and Electronic Engineering*, 11(2):178–184, 2016.

[111] L. Wang, S. Yu, Z. Wang, W. Qu, and H. Wang. Emotional classification of chinese idioms based on chinese idiom knowledge base. In *Workshop on Chinese Lexical Semantics*, pages 197–203. Springer, 2015.

[112] Z. Wang, V. J. C. Tong, and H. C. Chin. Enhancing machine-learning methods for sentiment classification of web data. In *Asia Information Retrieval Symposium*, pages 394–405. Springer, 2014.

[113] S. Wen and X. Wan. Emotion classification in microblog texts using class sequential rules. In *AAAI*, pages 187–193, 2014.

[114] M. Wöllmer, M. Kaiser, F. Eyben, B. Schuller, and G. Rigoll. Lstm-modeling of continuous emotions in an audiovisual affect recognition framework. *Image and Vision Computing*, 31(2):153–163, 2013.

[115] P.-k. Wong and C. Chan. Chinese word segmentation based on maximum matching and word binding force. In *Proceedings of the 16th conference on Computational linguistics-Volume 1*, pages 200–203. Association for Computational Linguistics, 1996.

[116] J. Woo, J. Botzheim, and N. Kubota. Facial and gestural expression generation for robot partners. In *2014 International Symposium on Micro-NanoMechatronics and Human Science (MHS)*, pages 1–6. IEEE, 2014.

[117] K. Xiao, Z. Zhang, and J. Wu. Chinese text sentiment analysis based on improved convolutional neural networks. In *2016 7th IEEE International Conference on Software Engineering and Service Science (ICSESS)*, pages 922–926. IEEE, 2016.

[118] C. Yang, K. H.-Y. Lin, and H.-H. Chen. Emotion classification using web blog corpora. In *IEEE/WIC/ACM International Conference on Web Intelligence*, pages 275–278. IEEE, 2007.

[119] M. Yang, W. Tu, J. Wang, F. Xu, and X. Chen. Attention based lstm for target dependent sentiment classification. In *AAAI*, pages 5013–5014, 2017.

[120] H. Zhang. The optimality of naive bayes. *AA*, 1(2):3, 2004.

[121] H.-P. Zhang, H.-K. Yu, D.-Y. Xiong, and Q. Liu. Hhmm-based chinese lexical analyzer ictclas. In *Proceedings of the second SIGHAN workshop on Chinese language processing-Volume 17*, pages 184–187. Association for Computational Linguistics, 2003.

[122] X. Zhang, J. Zhao, and Y. LeCun. Character-level convolutional networks for text classification. In *Advances in neural information processing systems*, pages 649–657, 2015.

[123] Y. Zhang, Y. Jiang, and Y. Tong. Study of sentiment classification for chinese microblog based on recurrent neural network. *Chinese Journal of Electronics*, 25(4):601–607, 2016.

[124] Y. Zhang, L. Shang, and X. Jia. Sentiment analysis on microblogging by integrating text and image features. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 52–63. Springer, 2015.

[125] J. Zhao, L. Dong, J. Wu, and K. Xu. Moodlens: an emoticon-based sentiment analysis system for chinese tweets. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1528–1531. ACM, 2012.

[126] C. Zhou, C. Sun, Z. Liu, and F. Lau. A c-lstm neural network for text classification. *arXiv preprint arXiv:1511.08630*, 2015.

[127] H. Zhou, M. Huang, T. Zhang, X. Zhu, and B. Liu. Emotional chatting machine: emotional conversation generation with internal and external memory. *arXiv preprint arXiv:1704.01074*, 2017.