Annotation and Detection of Emotion in Text-based Dialogue Systems with CNN

Jialiang Zhao, Qi Gao
1. School of Automation, Beijing Institute of Technology, Beijing 100081, P. R. China
E-mail: alanjzl@foxmail.com
E-mail: gaoqi@bit.edu.cn

Abstract: Knowledge of users’ emotion states helps improve human-computer interaction. In this work, we presented EmoNet, an emotion detector of Chinese daily dialogues based on deep convolutional neural networks. In order to maintain the original linguistic features, such as the order, commonly used methods like segmentation and keywords extraction were not adopted, instead we increased the depth of CNN and tried to let CNN learn inner linguistic relationships. Our main contribution is that we presented a new model and a new pipeline which can be used in multi-language environment to solve sentimental problems. Experimental results shows EmoNet has a great capacity in learning the emotion of dialogues and achieves a better result than other state of art detectors do.

Key Words: emotion detection, CNN, human-computer interaction, natural language processing

1 Introduction

Emotion detection is potentially important in voice recognition systems, which provides improvements to further human-computer interactions.

Emotion detection can be achieved by various approaches, including text, gestures, facial expression, electrocardiographs, etc[1-4]. Much work based on these features has been introduced over the last decade[5]. However, text is the most common way for communication in the Internet. Text-based emotion detection has multiple applications in human-computer interaction, especially in search engines and chatting robots. With the knowledge of users’ emotion, search engines could give more specific results, while chatting robots could reply in a more natural way.

Methods in text-based emotion detection can be divided into three categories: keyword-based detection, learning-based detection, and hybrid detection[5]. CNNs (convolutional neural networks) have shown their great capacity of learning the nature of images and speeches over the past few years. With a deep configuration and a large training dataset, keyword extraction might not be necessary anymore, which will significantly simplify the classification process.

In this work, we presented and tested a Chinese daily dialogues emotion detector based on deep convolutional neural network without segmentation or keyword extraction. This paper is divided into 7 sections. Section 2 gives an introduction of the pre-processing step. Section 3 gives the CNN configuration of EmoNet. Section 4 gives details of the training process. In section 5, we tested different configurations and hyper-parameters and gave evaluations based on top-1 accuracy and comparisons with other state of art detectors in different scales.

2 Dialogue Pre-processing

We manually labeled over 12,000 dialogues for training and evaluation. They are divided into 4 categories: positive, negative, wondering, and neutral. Most of them come from scripts of TV shows, movies and books. Numbers of dialogues in different categories are listed in Tab[1]

|                | Overall | Positive | Negative | Wondering | Neutral |
|----------------|---------|----------|----------|-----------|---------|
|                | 12186   | 3679     | 4205     | 1747      | 2555    |

2.1 Typical Routine

Before diving into the training process, normally several pre-processing steps should be done in order to improve overall efficiency. For English speech classification work, a typical routine is as follows:

- Spell check: A widely used library for spell check is PyEnchant. This step is optional.
- Stemming or Lemmatization[6]: Both of them are used to find the original form of words, e.g. ‘talk’ is the original form of ‘talking’ or ‘talked’.
- Case normalization: Convert input to uniform uppercase or lowercase. For example, “Today I’m happy.” might be converted to ”today i’m happy.”.

However, for Chinese classification or recognition system, a special step, segmentation, generally needs to be done. Normally researchers will build a LUT (look up table) which contains thousands of common used Chinese words before training, and then the frequency of appearance of each word in the target dialogue will be checked. A number of segmentation algorithms have been developed over the last decade[7-9]. The frequency vector is finally used as training materials.

One of the most appealing advantage with this method is that outputs of segmentation are of uniform length (equals to the length of pre-built LUT), so that during most classification algorithms, e.g. KNN and SVM, it can achieve a better results. Another important advantage is that segmentation with pre-selected keywords saves much time for researchers and im-
proves the efficiency. However, drawbacks with this algorithm shouldn’t be ignored.

- Over segmentation: Similar to image processing, over segmentation may significantly decrease the accuracy of our classifier. For example, 'not happy' may be segmented to 'not' and 'happy'. Obviously a single 'not' is meaningless but a 'happy' leads to a positive emotion, thus this sentence will be classified as positive because of over-segmentation, while the original emotion should be negative.

- Loss of linguistic features: Order of words is lost after segmentation. Same set of words with different orders sometimes gives absolutely opposite emotions. For example, as demonstrated in Fig.1 “He’s happy while I’m sad.”, and ”I’m happy while he’s sad” share a same vocabulary, but give opposite emotion.

2.2 Pre-processing in EmoNet

In order to avoid the problems listed above, we didn’t adopt segmentation in EmoNet. And because our net is tested in Chinese dialogues, stemming and lemmatization are not necessary.

2.2.1 Removal of Stop Words

Many words or expressions do not contribute to emotion or have same like-hood of occurring in those sentences not relevant to the target one. Such words are called stop words. In order to save storage space and improve classification efficiency, we need to remove them before further training. A list of Chinese stop words from Data Hall is used.

As shown in Fig.2 we searched for stop words in every source dialogue and removed them in this step. Notice that the order is kept.

2.2.2 Re-encode Input Characters

UTF-8 (8-bit Unicode Transformation Format) is a large and powerful set of characters which encompasses most of the world’s writing systems. With UTF-8, EmoNet gains the ability to perform in multi-language dialogue systems. Normally, one UTF-8 character is encoded with 1-6 byte, and in Chinese encoding, 3-byte UTF-8 is used.

However, because of UTF-8’s large range, most information in this encoding system is useless. And because of the great number of Chinese characters, it’s unnecessary to express each character precisely. The next step is to re-sample the encoding of input characters.

Firstly, because our net currently only contains Chinese, English and Arabic numerals, we can truncate UTF-8 to a new set that only contains characters of these vocabularies. Tab.2 gives encoding of Chinese characters, full-width English characters and full-width Arabic numerals in UTF-8.

| Alphabet | Chinese | Uppercase English |
|----------|---------|-------------------|
| UTF-8    | u 4e00 - u 9fa5 | u ff21 - u ff3a  |
| Num      | 20902   | 26                |
| Alphabet | Lowercase English | Numerals |
| UTF-8    | u ff41 - u ff5a  | u ff10 - u ff19  |
| Num      | 26       | 10                |

After this procedure, there are totally 20964 characters left, most of which are Chinese characters.
2.2.3 Encoding and Re-sampling

Unlike image processing system, encoding of characters is not "continuous", which means adjacent codes not necessary give similar meanings, while value in imaging systems gives continuous effect in hue, saturation, etc. Thus we don’t need to represent each character precisely. In EmoNet, we re-sample the 20964 codes into 256 codes with Eq. 1.

\[ \lambda_{new} = \lambda_{ori} \% 256 \]

After this procedure, every character is represented with one Byte. We set the maximum length of one dialogue to be 144 in EmoNet, thus totally \(144 \times 1 = 144\) Bytes are used to represent one dialogue.

3 Convolutional Neural Network in EmoNet

We adopted a deep CNN to do the classification job in EmoNet. CNN is a powerful tool with an adjustable capacity by adjusting its depth and breadth. With large datasets, CNN can achieve high performance in learning the nature of speech systems[16].

In EmoNet, we adopted a CNN which shares a similar but deeper configuration with LeNet-5[17], which consists of steps of convolutions, pooling, and fully-connected layers. Fig 3 gives the overall structure of EmoNet.

Fig. 3: Network Structure of EmoNet.

Input layer is a \(144 \times 1\) vector. The augmentation layer is used for feature expansion, which outputs a matrix of \(32 \times 32 \times 3\). Five convolutional layers are used, and most of them come with a max pooling layer for sub-sampling and dimension reduction. The last convolutional layer gives an output of \(6 \times 6 \times 256\). A fully-connected layer with dropout and softmax method is used for classification. The Output is of 4-dimension, positive, negative, wondering, and neutral.

Following sections give introduction of function and configuration of these layers in detail.

3.1 Augmentation Layer

We applied an augmentation layer to expand feature from the input and scale for the following layers. Because generally every convolutional layer outputs a matrix of smaller size but larger dimension compared with its input matrix, in order to apply enough number of convolutional layers, we need to expand the input matrix to a larger square matrix.

In every convolutional layer, we chose to use a filter of size \(5 \times 5\) and stride 1. In order to outputs a \(12 \times 12 \times n\) in the last step of convolutional step and apply 5 convolutional layers, we need an square output of size 32 in the augmentation step, which can be calculated with Eq. 2.

\[ \text{Size} = S_{output} + N_{layers} \times (S_{filter} - Stride) \]
\[ = 12 + 5 \times (5 - 1) \]
\[ = 32 \]

An affine layer follows Eq 3 where the weight matrix \(W\) is of size \([144, 32 \times 32 \times 3 = 3072]\), and bias vector \(B\) is of size \([3072, 1]\).

\[ \text{Output} = W \cdot \text{Input} + B \]

3.2 Convolutional and Pooling Layer

The convolutional part consists of 5 convolutional layers and 4 pooling layers.

3.2.1 Convolutional Layer

The convolutional layer adopts a filter to compute convolutions and generates a higher dimension matrix. In EmoNet, the filter sizes of all the 5 convolutional layers are chosen as \(5 \times 5\). Filter dimensions vary in different layers.

Typically, a sigmoid nonlinearity (or activation function), which is given in Eq 4 is used after every convolutional layer.
\[ S(x) = \frac{1}{1 + e^{-x}} \quad (4) \]

However, because sigmoid is easy to saturate, and according to Nair and Hinton, we use Rectified Linear Units (ReLUs) \[19\], a non-saturating nonlinearity instead. ReLUs follow Eq. 5.

\[ S(x) = \begin{cases} 
  x, & x \geq 0 \\
  0, & x < 0 
\end{cases} \quad (5) \]

3.2.2 Pooling Layer

Pooling layers introduce invariance, reduce dimension and prevent overfitting in CNNs \[18\].

We adopted overlapping pooling layers in this network. Overlapping pooling means adjacent pooling regions overlap with each other. For example, the pooling region size is denoted as \( s \times s \), and stride is denoted as \( z \). If \( s \geq z \), which is used in most CNNs, it is non-overlapping. If \( s < z \), then it becomes an overlapping pooling. According to Krizhevsky, Sutskever, and Hinton, an overlapping scheme will reduce error rate and decrease the probability of overfitting \[20\]. We adopted \( s = 5 \), \( z = 1 \) in EmoNet.

Also, many pooling methods have been developed, such as max pooling, average pooling, chunk pooling, etc. We used max pooling in EmoNet. Its principle is illustrated in Fig. 4.

Fig. 4: Maxpooling. Every \( 2 \times 2 \) block from the original matrix is pooled into a single block with the max value. Dimension of output matrix can be gained with \( D_{out} = \frac{D_{in}-(D_{filter}-Stride)}{Stride} \).

3.3 Fully Connected Neural Network with Dropout

3.3.1 Fully Connected Layers

A 3-layer fully connected neural network is connected to the last convolutional layer. Fully connected layers are affine layers, and every neuron in one layer is connected to each neuron in its adjacent layer. Each neuron of the first two layers comes with an RELU activation. Detailed configuration is listed in Table 3.

![Maxpooling with 2x2, stride 1](image)

Fig. 4: Maxpooling. Every \( 2 \times 2 \) block from the original matrix is pooled into a single block with the max value. Dimension of output matrix can be gained with \( D_{out} = \frac{D_{in}-(D_{filter}-Stride)}{Stride} \).

| Number of neurons | Layer 1 | Layer 2 | Layer 3 |
|-------------------|---------|---------|---------|
| 9216              | 1024    | 1024    | 1024    |

The final layer outputs a \( 5 \times 1 \) vector. This vector is then transformed into 5 categories, positive, negative, wondering, neutral and meaningless (which means this input should be discarded) by softmax classification.

3.3.2 Overfitting

Overfitting is a common problem in neural networks, especially when researchers don’t have a large enough dataset. Nowadays there are two popular methods to prevent overfitting, batch normalization and dropout.

We tested EmoNet with batch normalization first. During training, inputs of every layer are always changing with the fact that parameters in the prior layers are changing at the same time. This phenomenon, which is also called internal covariate shift \[21\], slows down the training process and requires careful initialization. Batch normalization solves this problem by normalizing layer inputs. However, in EmoNet, it reduced the error rate only by 2.5% to 3.8% compared with EmoNet without batch normalization with same amount of training. We think because our input is characters, which are discrete (while input of imaging systems are continuous), normalizing them might be unhelpful. Thus we tried dropout instead.

The core principle of dropout is to randomly disable some neurons, along with its connections during training \[22\], but do nothing during test. According to Srivastava et al., the optimal retention probability \( p \) of input layer should be close to 100% while with hidden layers it should be close to 50% \[22\]. We set \( p_{input} \) as 1.0 and \( p_{hidden} \) as 0.3, and we saw a reduction of error rate of around 6.5%.

4 Training Process

Parameters need to be updated during training, and normally people use back-propagation \[24\]. During this procedure, many parameters updating methods have been developed, such as SGD, momentum, RMSprop, and Adam. Adam is a first-order gradient-based optimization based on adaptive estimates of lower-order moments which can achieve a faster convergence than other methods like SGD, momentum and RMSprop \[25\].

We chose Adam as EmoNet’s parameter updating method. We divided the dataset into 2 parts, one of which was used for training, the other was used for evaluation. Cross validation \[26\] is used during the parameter updating process, and the training data was further divided into 32 mini-batch for cross validation.

5 Optimization

5.1 Test of Different Configurations

Configuration of EmoNet is modified from VGGNet \[23\]. We tested different configurations of EmoNet. Validation error and training speed within one epoch using same amount of data are used as evaluation criteria. Tab. 4 gives the configurations we tested, and Fig. 5 gives comparison between them. Matplotlib \[27\], a python plotting package, is used for drawing.
We chose configuration B, which achieved relatively high accuracy and fast performance at the same time, as the final structure of EmoNet.

5.2 Test of Different Parameters

Learning rate, regulation strength in affine layers, and mean of initialization are also important hyper-parameters that give huge influence on final results. Because dropout makes careful initialization less important, we only tested different sets of learning rate $\gamma$ and regulation strength $L$. Results are collected with same amount of input data and same number of training iterations. Results are shown in Fig. 6.

The group of highest accuracy, group $C$ is chosen, where $\gamma = 5 \times 10^{-6}, L = 1.5 \times 10^{-4}$. The loss v.s. time curve of the first epoch is shown in Fig. 7.

5.3 Evaluation

With optimized parameters, EmoNet was trained with tensorflow and Nvidia GPU. After 100 epochs’ training, an overall top-1 accuracy of 72.8% was achieved. Tab. 5 gives accuracy of each category separately.

According to the results, neutral and wondering states are relatively easier to detect than neutral and negative states. Wondering states are the easiest to detect, which is also easy to interpret. Generally dialogues of this emotion state come with question marks, or some specific words, like “what”, “why”, “how”, etc.

Table 4: Configurations of EmoNet

| No. | A          | B          | C          | D          |
|-----|------------|------------|------------|------------|
| No. |            |            |            |            |
| A   | 9 weighted layers | 9 weighted layers | 9 weighted layers | 9 weighted layers |
| B   | Conv32     | Conv32     | Conv32     | Conv32     |
|     | Conv32     | Conv32     | Conv32     | Conv32     |
| C   | Conv32     | Conv32     | Conv64     | Conv64     |
|     | Conv64     | Conv64     | Conv64     | Conv64     |
| D   | Conv128    | Conv128    | Conv128    | Conv128    |
|     | Conv128    | Conv128    | Conv256    | Conv256    |
|     | Conv256    | Conv256    | Conv256    | Conv256    |
|     | Conv256    | Conv256    | Conv256    | Conv256    |
|     | Maxpool 5 × 5, stride 1 | Maxpool 5 × 5, stride 1 | Maxpool 5 × 5, stride 1 | Maxpool 5 × 5, stride 1 |
|     | Dropout    | Dropout    | Dropout    | Dropout    |
|     | Fully connected layer 1024 | Fully connected layer 1024 | Fully connected layer 1024 | Fully connected layer 5 |
|     | Softmax    | Softmax    | Softmax    | Softmax    |
Although accuracy of neutral state is high, we found that most emotion states of false detected dialogues were detected as neutral. This phenomenon may result from two facts, the first one is that a neutral state is too vague and general, and the other one is that unlike other states in which we can find specific emotional words (for example, "why" for wondering states and "happy" for positive states), there are no specific words for neutral states.

Detection with positive states and negative states are similar. However, as we can see, accuracy with positive states is low. This may result from the quality of the training materials. Ironies are very common in our source, and it is sometimes even hard for us to distinguish whether a dialogue is positive or negative.

Comparisons between EmoNet and state of art Chinese text-based emotion detectors are given in Tab[6] Tab[7] and Tab[8]. Multi-Model Net was given by Ze-Jing Chuang and Chung-Hsien Wu[28], and ESiN was given by Jianhua Tao[29].

6 Conclusion

In this paper, EmoNet was presented, an emotion detection system based on deep convolutional neural networks. We analyzed the feature of Chinese encoding and adopted a preprocessing step without segmentation, stemming or lemmatization, which introduces difficulties but address the problem of loss of linguistic features. Now in EmoNet, a simple re-sampling step was used to replace these steps. In the future, we will try some other algorithms. We are now working in developing a linear mapping system which can map arbitrary-length dialogues into equal-length outputs.

Different CNN configurations and different hyper-parameters were tested. An overall accuracy of 0.72 was achieved with 12,000 training dialogues, 100 epochs’ training and the optimized modal. Top-1 accuracy of EmoNet is higher than other Chinese text based emotion detector. However, it’s foreseeable that with more training materials, EmoNet has the capacity to achieve a better performance.

References

[1] Cohn J F, Katz G S. Bimodal expression of emotion by face and voice[C]//Proceedings of the sixth ACM international conference on Multimedia: Face/gesture recognition and their applications. ACM, 1998: 41-44.
[2] Devillers L, Lamel L, Vasilescu I. Emotion detection in task-oriented spoken dialogues[C]//Multimedia and Expo, 2003. ICME’03. Proceedings. 2003 International Conference on. IEEE, 2003, 3: III-549.

[3] Agrafioti F, Hatzinakos D, Anderson A K. ECG pattern analysis for emotion detection[J]. IEEE Transactions on Affective Computing, 2012, 3(1): 102-115.

[4] Gunes H, Piccardi M. Bi-modal emotion recognition from expressive face and body gestures[J]. Journal of Network and Computer Applications, 2007, 30(4): 1334-1345.

[5] Tinghao Y, Hsieh C T, Soo V W. Towards Text-based Emotion Detection[C]//Proc. of International Conference on Information Management and Engineering.[S. l.]: IEEE Press. 2009: 70-74.

[6] Korenius T, Laurikkala J, Järvelin K, et al. Stemming and lemmatization in the clustering of finnish text documents[C]//Proceedings of the thirteenth ACM international conference on Information and knowledge management. ACM, 2004: 625-633.

[7] Liu K Y, Zheng J H. Research of automatic Chinese word segmentation[C]//Machine Learning and Cybernetics, 2002. Proceedings. 2002 International Conference on. IEEE, 2002, 2: 805-809.

[8] Peng F, Feng F, McCallum A. Chinese segmentation and new word detection using conditional random fields[C]//Proceedings of the 20th international conference on Computational Linguistics. Association for Computational Linguistics, 2004: 562.

[9] Sproat R, Gale W, Shih C, et al. A stochastic finite-state word-segmentation algorithm for Chinese[J]. Computational linguistics, 1996, 22(3): 377-404.

[10] Huang C, Zhao H. Chinese word segmentation: A decade review[J]. Journal of Chinese Information Processing, 2007, 21(3): 8-20.

[11] Patino L. Fuzzy relations applied to minimize over segmentation in watershed algorithms[J]. Pattern Recognition Letters, 2005, 26(6): 819-828.

[12] Wilbur W J, Sirotkin K. The automatic identification of stop words[J]. Journal of information science, 1992, 18(1): 45-55.

[13] Hao L, Hao L. Automatic identification of stop words in Chinese text classification[C]//Computer Science and Software Engineering, 2008 International Conference on. IEEE, 2008, 1: 718-722.

[14] Data Hall, List of Chinese Stop Words[J]. 2016-07-05. [http://www.datatang.com/data/193007/Data Hall. Stop Words Set.

[15] Yergeau F. UTF-8, a transformation format of Unicode and ISO 10646[R]. 1996.

[16] Krizhevsky A, Sutskever I, Hinton G E. Imagenet classification with deep convolutional neural networks[C]//Advances in neural information processing systems. 2012: 1097-1105.

[17] LeCun Y, Jackel L D, Bottou L, et al. Learning algorithms for classification: A comparison on handwritten digit recognition[J]. Neural networks: the statistical mechanics perspective, 1995, 261: 276.

[18] Scherer D, Miller A, Behnke S. Evaluation of pooling operations in convolutional architectures for object recognition[J]. Artificial Neural Networks-CICANN 2010, 2010: 92-101.

[19] Nair V, Hinton G E. Rectified linear units improve restricted boltzmann machines[C]//Proceedings of the 27th international conference on machine learning (ICML-10). 2010: 807-814.

[20] Krizhevsky A, Sutskever I, Hinton G E. Imagenet classification with deep convolutional neural networks[C]//Advances in neural information processing systems. 2012: 1097-1105.

[21] Ioffe S, Szegedy C. Batch normalization: Accelerating deep network training by reducing internal covariate shift[C]//International Conference on Machine Learning. 2015: 448-456.

[22] Srivastava N, Hinton G E, Krizhevsky A, et al. Dropout: a simple way to prevent neural networks from overfitting[J]. Journal of Machine Learning Research, 2014, 15(1): 1929-1958.

[23] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition[J]. arXiv preprint arXiv:1409.1556 2014.

[24] Hecht-Nielsen R. Theory of the backpropagation neural network[J]. Neural Networks, 1988, 1(Supplement-1): 445-448.

[25] Kingma D, Ba J. Adam: A method for stochastic optimization[J]. arXiv preprint [arXiv:1412.6980] 2014.

[26] Kohavi R. A study of cross-validation and bootstrap for accuracy estimation and model selection[C]//Ijcai. 1995, 14(2): 1137-1145.

[27] Barrett P, Hunter J, Miller J T, et al. matplotlib–A Portable Python Plotting Package[C]//Astronomical Data Analysis Software and Systems XIV. 2005, 347: 91.

[28] Chuang Z J, Wu C H. Multi-modal emotion recognition from speech and text[J]. Computational Linguistics and Chinese Language Processing, 2004, 9(2): 45-62.

[29] Tao J. Context based emotion detection from text input[C]//Interspeech. 2004.