NLPSA at IJCNLP-2017 Task 2: Imagine Scenario: Leveraging Supportive Images for Dimensional Sentiment Analysis

Szu-Min Chen, Zi-Yuan Chen, Lun-Wei Ku

Academia Sinica 128 Academia Road, Section2 Nankang, Taipei 11529, Taiwan {b02902026, b02902017}@ntu.edu.tw lwku@iis.sinica.edu.tw

Abstract

Categorical sentiment classification has drawn much attention in the field of NLP. while less work has been conducted for dimensional sentiment analysis (DSA). Recent works for DSA utilize either word embedding, knowledge base features, or bilingual language resources. In this paper, we propose our model for IJCNLP 2017 Dimensional Sentiment Analysis for Chinese Phrases shared task. Our model incorporates word embedding as well as image features, attempting to simulate human's imaging behavior toward sentiment analysis. Though the performance is not comparable to others in the end, we conduct several experiments with possible reasons discussed, and analyze the drawbacks of our model.

1 Introduction

Dimensional Sentiment prediction is a subcategory of sentiment analysis. Traditionally, the goal of the sentiment classification task is either binary, mostly positive and negative, or categorical, such as happy, angry, and sad(Pang et al., 2002; Rosenthal et al., 2017). Instead of categorizing different emotions to a fixed number of classes, the dimensional approach projects each emotion to valencearousal (VA) space. Valence indicates the level of pleasant and unpleasant, while arousal shows the level of excitement and calm. This methodology has drawn more attention recently since the valence-arousal space is continuous comparing to the discrete classes used previously, so it implies the better capability to describe the emotion, while better benefiting downstream model for further application.

In this paper we propose our model participating in the IJCNLP 2017 dimensional sentiment analysis shared task in which participants are asked to predict the numerical valence-arousal on words and phrases. For details of the shared task, 2,802 words are given for training and 750 words are for testing; 2,250 and 750 phrases are provided in the phrase track, for training and testing respectively. Phrases are composed by at least one of the degree, negation or modal with a word. Degree stands the level for the feeling, negation means the negative of the feeling, and modal represents the frequency of the feeling. Examples can be found in Table 2. And the type of modifier of each phrase is given, as shown in Table 3. The predict result is evaluated by the organizer with the mean absolute error (MAE) and the Pearson correlation coefficient (PCC) (Pearson, 1895) as metrics.

Recent approaches tackling the VA prediction problem on word and phrase includes neural-network based method and graphical-based method, with detail described in Section 3. However, the former suffers from the insufficient data, especially data in Chinese, while the latter heavily utilizes pretrained word-embedding which mostly capture syntactic features of words but not semantic features.

We therefore consider the way how human will determine sentiment of given words or phrases. In our experience, humans imagine the scene when they are given such word or phrase. For example, when humans heard the word "欣喜若狂"(ecstatic), they will think of acclamation, laughter on faces, and other positive scenes, as shown in Figure 1. These imaginations can be regarded as contexts of the word "欣喜若狂"(ecstatic), which helps humans decide the degree of positive/negative and exciting/calm of the word.

Considering the data inadequacy of annotated



Figure 1: The example of scene that people may imagine with the word "欣喜若狂"(ecstatic). These images are directly downloaded from the Internet.

Word			
Word	Valence	Arousal	
成功(success)	8.2	6.6	
暗殺(assassinate)	1.8	6	
狂喜(ecstatic)	8.6	8.8	
Phrase			
Phrase	Valence	Arousal	
最為出色most excellent)	8.056	6.288	
極度失望(very disappoint)	1.63	7.244	
略為放鬆(little relaxing)	6.2	1.2	

Table 1: Examples of word and phrase training instances.

VA corpus and the behavior of human, we proposed a model that takes the word as well as the images related to that word, which are directly downloaded from internet, as input. The key idea is to leverage the huge amount of information on the Internet and the feature extraction ability of convolution neural network to overcome the small size of annotated datas, and also balance the importance of word embedding with respect to the view from model.

The rest of this paper is organized as following: Section 2 overviews the shared task, related works are reviews in Section 3, Section 4 illustrates our valence-arousal prediction with image model, Section 5 is for the experiment and the result, the experimental results are discussed and analyzed in Section 6, and Section 7 concludes the paper with future works.

2 Related Work

There are several different methods toward the VA prediction task, some researches about the valence arousal prediction on paragraphs are conducted (Wang et al., 2016b; Nicolaou et al., 2011), while

degree	negation	modal
十分(really)	不能(cannot)	也許(maybe)
稍微(little)	沒(no)	本來(originally)
超級(very)	沒有(no)	應該(perhaps)

 Table 2: Examples of degree, negation, modifier

 in phrase

modifier's type	phrase
deg	異常驚人
neg_deg	沒有太驚訝
mod_neg	本來不支持

Table 3: Examples of types of modifiers and corresponding phrases. "deg", "neg", "mod" stand for "degree", "negation", "modal" respectively. Besides, the "_" symbol denotes combinations of modifiers in order. The three phrases in second column represent "extraordinarily surprising", "not too surprised", and "was not supported originally" in English, respectively.

in this work, our goal is to directly predict the values of valence and arousal on words and phrases only. Thus, we will discuss those works which utilize the words and some kinds of auxiliary information, like the translation from English to Chinese or word tag from Wordnet, but do not incorporate the context information, like a sentence annotated with VA score. Also, we will briefly cover some works corresponding to image sentiment classification.

2.1 Graph-Based Approach

In the work of (Yu et al., 2015), the objective is to predict the V and A score for each single word. The method proposed is a weighted graph model, each vertex is a vector representation of a word, and the edge is weighted by the cosine similarity between two words. The unseen words use the neighboring seed words' labeled scores, which weighted by the similarity, to update themselves' score. This can be considered an iterative process and will loop until converge, and since the similarity is calculated based on word embedding, it is critical that the word embedding used in training can indeed represent the relationship between word with respect to sentiment. In the work of (Wang et al., 2016a), they try to deal with the task by purposing a community-based method. They define a Modularity term which can be viewed as the difference between the sum of all similarities between words within a community C, and the

sum of all similarities between words in C and all the other words that not belongs to C. The training step is to maximize the modularity term with respect to all communities. After convergence they treat each community as a new graph and predict unseen words' rating from their neighbors. A main drawback of graph-based method is that they almost depend solely on the information provided by word embedding which contain inadequate semantic features, thus make the model cannot be interpreted intuitively.

2.2 Linear-Regression Approach

Besides graphical model, some of the other works use linear regression to deal with the problem as in (Wei et al., 2011) and (Wang et al., 2015). Both of the works use English corpus (ANEW) as source domain and transform the rating to the Chinese words. The former work first cluster both English and Chinese work into several groups based on the SUMO concept in WordNet, then for each different cluster they use a linear regression model to predict Chinese words' VA score based on the corresponding English words. The latter work use a variation of linear regression, where a datapoint will be locally weighted by its neighbors' VA score. Also we found that the interrelationship between valence and arousal ratings is considered in this work, and they use quadratic polynominal function to model the relationship and use it as the regression features.

2.3 Image Sentiment Classification

Though object or facial detection of image has made lots of progress in recent years, we notice that, as far as we know, only few works try to deal with sentiment classification in image domain. Reasons might be the lack of annotated data and the difficulty of this task, which need to determine the emotion by taking all the information in the picture into consideration, such as the facial expressions or color tone of the picture that related to some higher-level abstract concept, rather than merely find some specific objects. The work in (You et al., 2015) train a convolution neural network model on a weakly-labeled (where labels are machine generated) Flicker dataset. It uses a progressive training method which first makes prediction then selects a subset with higher prediction confidence in term of probability in the training data, then use them to further fine-tuned the model. The experiment was also be conducted that

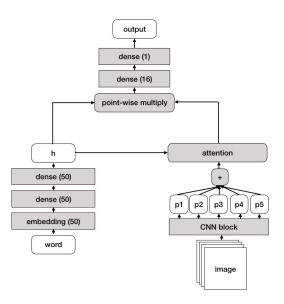


Figure 2: Model for dimensional sentiment analysis

transfer the model from Flicker dataset to Twitter and yield promising performance, thus we think convolution neural net indeed can extract some sentiment features.

3 Dimensional Sentiment Analysis by Image

This section introduces the model architecture. Section 4.1 propose our model for word track, and Section 4.2 illustrates our model for phrase track. Settings of hyperparameters are defined in Section 6.1.

3.1 Word-Level

The model we proposed is in figure 2. For each word in training set, we will choose five images gathered from the internet as input as well as that word. The model has two separate parts for processing two different kinds of input.

3.1.1 Image Processing

The right part of the model is for processing images. There are six convolution layers totally, while each two layers can be viewed as a block since they have same filter sizes and filter numbers. At the end of each block there is a maxpooling layer. After the CNN there is a fullyconnected layer used to extract the features from images. Then we use another dense layer to generate a scalar, which indicates the polarity of the word, from each of five vectors. The scalars are denoted as $P_j \in \mathbb{R}^k$, where k is the dimension of word embedding, $j \in \{0, 1, 2, 3, 4\}$ means the j^{th} image

3.1.2 Word Processing

The left part of the model is used to process word information. We use Glove as pretrained word embedding and freeze the weight during training phase. Then we use two fully connected layers, by which we hope to extract sentiment features from original word embedding, the output can be viewed as sentiment word vector, denoted as h_i .

3.1.3 Integration of word and images

After processing two kinds of inputs, we apply the attention mechanism on five polarity scalars with respect to the sentiment word vectors. The calculation of attention weights can be written as

$$e_{i,j} = \tanh\left(W_a h_i + W_b P_{ij} + b\right) \tag{1}$$

$$a_{i,j} = \frac{exp(e_{i,j})}{\sum_{j'=0}^{4} exp(e_{i,j'})}$$
(2)

where $W_a \in \mathbb{R}^{1 \times k}, W_b \in \mathbb{R}^{1 \times k}, b \in \mathbb{R}$ are learned by the model during training. After the calculation of the attention weight, we perform element-wise multiplication on the polarity vectors obtained from images.

$$c_i = \sum_{j'=0}^{4} a_{i,j'} P_i$$
 (3)

Obtaining the weighted polarity vector c_i , we then perform element-wise multiplication on sentiment word vector and the output vector of attention mechanism to calculate

$$h_i' = c_i \otimes h_i \tag{4}$$

Since the images are collected from the Internet, we have no guarantee about the qualities of the images. Some images, in consequence, might be useless or even have negative effect for the model to predict valence or arousal score. The attention mechanism above gives our model the capacity to determine how importance each image is with respect to reflecting the sentiment. For example, it can give lower attention weight to images regarded as irrelevant to the proceeding words, then using less feature from the images in final prediction stage.

The last part of the model is a fully-connected layer with *sigmoid* activation function, of which output is a real-value prediction.

3.2 Phrase

We model a phrase as a word with modifiers, thus we add several layers to deal with modifiers explicitly and leave other parts of model unchanged.

Since the modifiers' types are given, an intuitive method to process phrase is to multiply the representation of those modifiers and word. The reason is that the modifiers will change the meaning and/or intensity of a word, which can be explained as the alteration of a vector in word embedding space.

A concrete example is to represent a negation, we could multiply -1 on it. Multiplying -1makes opposite direction of any vectors, so it gives the opposite meaning of word vectors. Note that since we are using same embedding for modifiers and words, they are in the same embedding space. So the direct multiplication of the modifier embedding and the word embedding is achievable and reasonable.

Given a phrase, we first split it into degree, negation, modal and word, while a matrix fulled with one is used if such modifier does not exist. A same pretrained word embedding is used to process them to the same embedding space. With the embedding, a dense regression is performed on the degree embedding, negation embedding, and the modal embedding, while a different dense regression is used for the word embedding. In the end, we multiply the vectors to get the representation of phrase.

With the same process attention mechanism performed on images, we then multiply the representation of phrase we obtain above with the polarity from images. Then a fully-connected neural network and a *sigmoid* function is used to compute the final output.

4 Experiment

In this section, we describe the setting of model we use on shared task. Furthermore, we conduct two additional experiment as below, to analyze the behavior of our model.

1. Using different numbers of images for each word and comparing the performance.

Submit	Valence MAE	Valence PCC	Arousal MAE	Arousal PCC	
	Word Level				
Run1	1.108 (22)	0.561 (22)	1.207 (23)	0.351 (22)	
Run2	1 (19)	0.604 (22)	1.207 (23)	0.351 (22)	
Phrase Level					
Run1	0.709 (14)	0.818 (15)	0.632 (16)	0.732 (16)	
Run2	0.689 (13)	0.829 (14)	0.633 (17)	0.727 (17)	

Table 4: Result of IJCNLP 2017 dimensional sentiment analysis shared task. Rank among all submission is shown in the bracket.

2. Comparing the model with the other neuralnetwork based model which does not incorporate the image information

4.1 Model Settings

In the word-level experiment, we use 50dimension GloVe which trained on Chinese Gigaword Corpus as pretrained word embedding, 3 CNN blocks use 16, 32, 64 as filter numbers respectively and 3x3 as kernel size. Following the CNN we use a fully-connected layer with 200 hidden units. On the other side of the model, a two layers dense with hidden size 50 is used for regression. In the end, there is a dense layer with hidden size 16 for final computation.

About training, we use mean squared error as our loss function, mini-batch with size 16, and use Nadam(Dozat, 2016) for optimization, learning rate is initialized to 0.005 and 0.004 scheduled decay. Dropout layers (Srivastava et al., 2014) with dropout rate 0.3 and batch normalization mechanism (Ioffe and Szegedy, 2015) are inserted after each CNN block, while early-stopping being used with monitoring the behavior of validation mean absolute error since it is the metric used in shared task. We shuffle and partition the data to train on 2661 instances and validation on 141 instances. Hoping to lower the variance in the training data, we manually divide the VA rating with 10 and multiply it back to original scale after prediction. At each epoch we save the best model with respect to validation mean absolute error and use it to evaluate the current Pearson Correlation Coefficient.

Only slightly different settings are applied to phrase-level experiment. There are two regressions for modifiers and words, each with hidden size 50, and there is no batch normalization mechanism after each CNN block. After shuffle and the same partition strategy, we have 2025 training instances and 225 validation instances. In the end,



Figure 3: Most of images show no relevance to the corresponding word and phrase.

a same methodology to prevent high variance is adapted.

We submit two runs to the system. In the second run we fine-tuned the hyper-parameter of the phrase model and we use over-sampling to train the word model since we find there are few instances with extreme ratings. So for those datapoints whose ratings are very large (> 7) or small (< 3), we replicate them two times in training set.

4.2 Additional Experiment

The details of two additional experiments are introduced as following.

Using different numbers of images Motivation of our first experiment is that the images we used in our models are automatically downloaded from the Internet, and we have no guarantee about relevance between images and corresponding words or phrases. Therefore we try different numbers of images for training, trying to implicitly control the degree our model relies on image information.

Comparing to an FFNN model To evaluate the usefulness of images, we also implement a feed-forward neural network (FFNN) containing two hidden layers that directly takes word embedding as input and output the valence or arousal rating of each word.

number of images	Valence MAE	Valence PCC	Arousal MAE	Arousal PCC
5 images	0.618(1.15)	0.838(0.55)	0.845(1.15)	0.54(0.37)
4 images	0.718(1.07)	0.74(0.59)	0.811(1.054)	0.60(0.39)
3 images	0.792(1.01)	0.75(0.60)	0.868(1.072)	0.57(0.37)
2 images	0.679(1.06)	0.82(0.53)	0.826(1.063)	0.53(0.35)

Table 5: The relationship between number of images and performance in validation set, while the value in the bracket denotes the performance on testing set.

Model	Valence MAE	Valence PCC	Arousal MAE	Arousal PCC
FFNN	0.746(1.108)	0.79(0.57)	0.780(1.13)	0.58(0.35)
Our model	0.618(1.15)	0.838(0.55)	0.845(1.15)	0.54(0.37)

Table 6: Two model's behavior on the validation and testing set, our model using 5 images as input.

5 Results & Discussion

Results of shared task The shared task results of word- and phrase-track are showed in Table 4. The rank of each team is decided by the average of rank considering each metric, which is shown between brackets. In word-level, our models show relatively unsatisfactory performance among other teams, with rank 17.5/24 in first run and 17/24 in second run. We discover that the testing result is significantly worse than training and validation, and since the distribution of VA ratings in training and testing are similar after analysis, the possible reason is that the training set is too small for model that it's ability of generalize to unseen data is weak.

Result of phrase-level is better than that of word-level. We think the reason is that when our model encounter different phrase with some or all the same modifiers, the same modifiers can stand as a reference. This reference helps our model preciously update its representation of the different part of the phrase, which results in better performance.

Using different numbers of images The experiment result is shown in Table 6. We find that there is no obvious difference of the performance no matter how many images we utilize. By looking to Figure 4, the fourth row shows that the model is able to attend relevant images, but in second row, all the images are irrelevant so the model doesn't know what to do and give all the image and word embedding same attention weights. This indicates that the attention mechanism is not mature enough to ignore the useless images. The lack of training data is one reason and the quality of images also largely affects the model. **Comparing to an FFNN model** The result is shown in Table 7. According to this experiment, we notice that though FFNN model is quite unstable and the mean absolute error varies over multiple training phases, the model with image perform only slightly better than FFNN, sometimes even worse. After some investigation we conclude that this result is due to the poor relevance between images and words. So we further analyze the image data.

Quality of images We manually check lots of images and find that a considerable portion of the images crawled from internet are very irrelevant to the corresponding word and thus have negative effect, as shown in Figure 3. We use Google search due to the lack of image dataset with Chinese label. However, since the Google image search will consider the relation between a query and the paragraph that co-occur with each image, so we might download lots of images have little or no relevance to the word, and since many words in the dataset have abstract meaning, e.g. happy or sad, rather than just represent an object, e.g. table or cat, so it is much harder to get the images which represent the words accurately. For example, the first two images of the word "生育"(fertility) is appeared in article discusses the fertility policy of Nazi Germany, while the images of the word "誠 實"(honesty) are some images and posters of the movie "Gone Girl". "誠實"(honesty) is indeed relevant to the plot of the movie "Gone Girl", but the images of that movie are useless to our model. These pictures can serve as a distraction to the model rather than provide useful information.

Possible solutions Since the result of the shared task is unsatisfactory to us, we're going to point



Figure 4: The attention heat map of words and images shows that it cannot attend on words only, while often attend on irrelevant images.

out some problems and propose some possible solution. For the severe problem of image mentioned above, maybe we can use some image dataset with English label such as ImageNet since it is much bigger, then translate the Chinese word to English to obtain corresponding pictures and using Google search for auxiliary. Another reason should account for the bad performance is the lack of training data, it is hard to train a large network by using less than three thousand instances. To tackle this problem we can also incorporate English corpus with VA ratings to pretrain our model, or train a network for sentiment classifier first and transfer the weight to initialize our model.

6 Conclusion

In this paper, we propose a novel approach to tackle the dimensional sentiment analysis by incorporating information from images, a resource that, as far as we know, hasn't been used in this domain. Although the result isn't promising enough, we conduct several experiments and some feasible solutions are proposed, hoping to better utilize our model.

References

- Timothy Dozat. 2016. Incorporating nesterov momentum into adam.
- Sergey Ioffe and Christian Szegedy. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, pages 448–456.
- Mihalis A Nicolaou, Hatice Gunes, and Maja Pantic. 2011. Continuous prediction of spontaneous affect from multiple cues and modalities in valence-

arousal space. *IEEE Transactions on Affective Computing*, 2(2):92–105.

- Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, pages 79–86. Association for Computational Linguistics.
- Karl Pearson. 1895. Note on regression and inheritance in the case of two parents. *Proceedings of the Royal Society of London*, 58:240–242.
- Sara Rosenthal, Noura Farra, and Preslav Nakov. 2017. Semeval-2017 task 4: Sentiment analysis in twitter. In Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017), pages 502–518.
- Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *Journal of machine learning research*, 15(1):1929–1958.
- Jin Wang, Liang-Chih Yu, K Robert Lai, and Xue-jie Zhang. 2015. A locally weighted method to improve linear regression for lexical-based valence-arousal prediction. In Affective Computing and Intelligent Interaction (ACII), 2015 International Conference on, pages 415–420. IEEE.
- Jin Wang, Liang-Chih Yu, K Robert Lai, and Xuejie Zhang. 2016a. Community-based weighted graph model for valence-arousal prediction of affective words. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 24(11):1957–1968.
- Jin Wang, Liang-Chih Yu, K Robert Lai, and Xuejie Zhang. 2016b. Dimensional sentiment analysis using a regional cnn-lstm model.
- Wen-Li Wei, Chung-Hsien Wu, and Jen-Chun Lin. 2011. A regression approach to affective rating of chinese words from anew. *Affective Computing and Intelligent Interaction*, pages 121–131.
- Quanzeng You, Jiebo Luo, Hailin Jin, and Jianchao Yang. 2015. Robust image sentiment analysis using progressively trained and domain transferred deep networks. In *AAAI*, pages 381–388.
- Liang-Chih Yu, Jin Wang, K Robert Lai, and Xue-jie Zhang. 2015. Predicting valence-arousal ratings of words using a weighted graph method. In ACL (2), pages 788–793.