Script Character Emotion Recognition

Jianhua Tu (A20480216) Biao Sun (A20475197)

Ye Yu (A20478640)
Illinois Institute of Technology
{jtu5,bsun27,yy99}@hawk.iit.edu

Abstract

This task is to analyze and identify the emotions of each character involved in every dialogue and action description in the script scenes from multiple dimensions. Comparing with traditional sentimental classification task, this task has its own characteristics and challenges. Emotions are multidimensional including 6 different types: love,joy,anger,surprise,fear,sorrow. And each emotion has a degree. For example, the degree of happiness ranges from 0 to 3, with 0 being none and 3 being the strongest. Emotion classification is for a certain role in a sentence, rather than the whole sentence. A sentence may have multiple characters. Considering the property of the task, we tried quite a few methods. We use the existing python package simpletransformers multilableclassifier to implement the baseline model and build 3 improved models with Pytorch. The experiment result shows that our improved models overcome the limitations of the existing package and makes significant improvement in the RMSE score comparing to the baseline model.Our code is available at github. [1]

[1] https://github.com/tujianhua/script_emotion

1. Introduction

Since early 2000, sentiment analysis has grown to be one of the most active research areas in NLP, because people's opinions are central to almost all human activities and behaviours, hence sentiment analysis is very import to business and society. Sentiment analysis is usually formulated as a multi-category classification problem, namely, to predict a text as positive, neural, and negative, or binary classification problem, positive and negative.There are also some studies on fine-grained emotions or sometimes called multi-label emotion analysis([5]), and some dataset on the fine-grained emotions are published([1]). The difference between multi-label classification and multi-label classification lies in:

1. Multi-label classification refers to the classification of multiple aspects or targets of a sample, usually for each aspect it is binary classification, having only two categories. These aspects are not mutually exclusive. For example, for a sample image of a person, to classify into male or female in terms of gender, adult or child in terms of age. These aspects of the image can occur simultaneously,since a person has both gender and age properties.

2. Multi-classification problem is to tell which

category a sample belongs to, the value of category is more than 2. These categories are mutually exclusive meaning that if a sample belongs to category 1, it cannot belong to category 2 or category 3. For example, the task of face recognition is to classify faces into different people's faces, if it is person A's face, it couldn't be person B's face in the mean time.

1.1. Problem description

Compared with traditional sentiment analysis,an on-going competition task[2], the script character emotion recognition is a new area of research and is more challenging. The importance of script to film and television industry is self-evident. A good script is not only the basis of good word of mouth and traffic, but also can bring higher commercial returns. Script analysis is the first link in the production chain of film and television content, in which the emotion identification of script characters is a very important task, which is mainly to analyze and identify the emotions of each character involved in every dialogue and action description in the script from multiple dimensions. Compared with the usual news and commentary text sentiment analysis, it has its unique business characteristics and challenges. Usual news and commentary text sentiment analysis is to predict the polarity which usually has two classes:positive,negative or three classes: positive, neutral and negative, and this is a binary classification or 3 class multi-classification problem. While this task is much more complicated, it requires identify the emotion for different characters or roles mentioned in the script text, and furthermore, it requires to identify the emotion type and intensity degree of each emotion. The emotion depends on not only current text but also on the historic texts.

[2]https://www.datafountain.cn/competitions/518

1.2. Performance measure metric

The performance of the algorithm is measured by the common used root mean square error. Error is calculated according to every emotion value identified by the combination of text content plus character name.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}\sum_{j=1}^{6}(y_{i,j}-x_{i,j})^2}{6n}}$$

score = 1/(1 + RMSE)

Where $y_{i,j}$ is the predicted emotion value for the ith data sample and jth emotion type, $x_{i,j}$ are the labeled emotion value for the ith data sample and jth emotion type, and n is the total number of test samples. The final ranking is based on score.

2. Related work

Existing research has produced numerous techniques for various tasks of sentiment analysis, which include SVM, Maximum Entropy,Naïve Bayes and Neural Network([6]). The features of the text are very important for sentiment analysis. There are many works([3]) on how to explore the features in order to improve the performance. Bag of word, CounterVecttor TF-IDF, emotion vocabulary, special emotional lexicon are frequently used.

In recent years, Bert and its variants produced state-of-the-art results in many NLP application including the sentence pair classification task,single sentence classification task, question answering task, sequence labeling task. Using pre-trained BERT embedding for a single text or texts pair and fine-tuning it in a specific task becomes popular in NLP, because it transfers the general knowledge learnt in large scale corpus into the specific task and often gets good performance. BERT is based on transformer([4]) architecture whose major idea is adopting attention mechanism. The paper of BERT ([2]) provides the usage of BERT for the downstream tasks as shown in Figure 1.

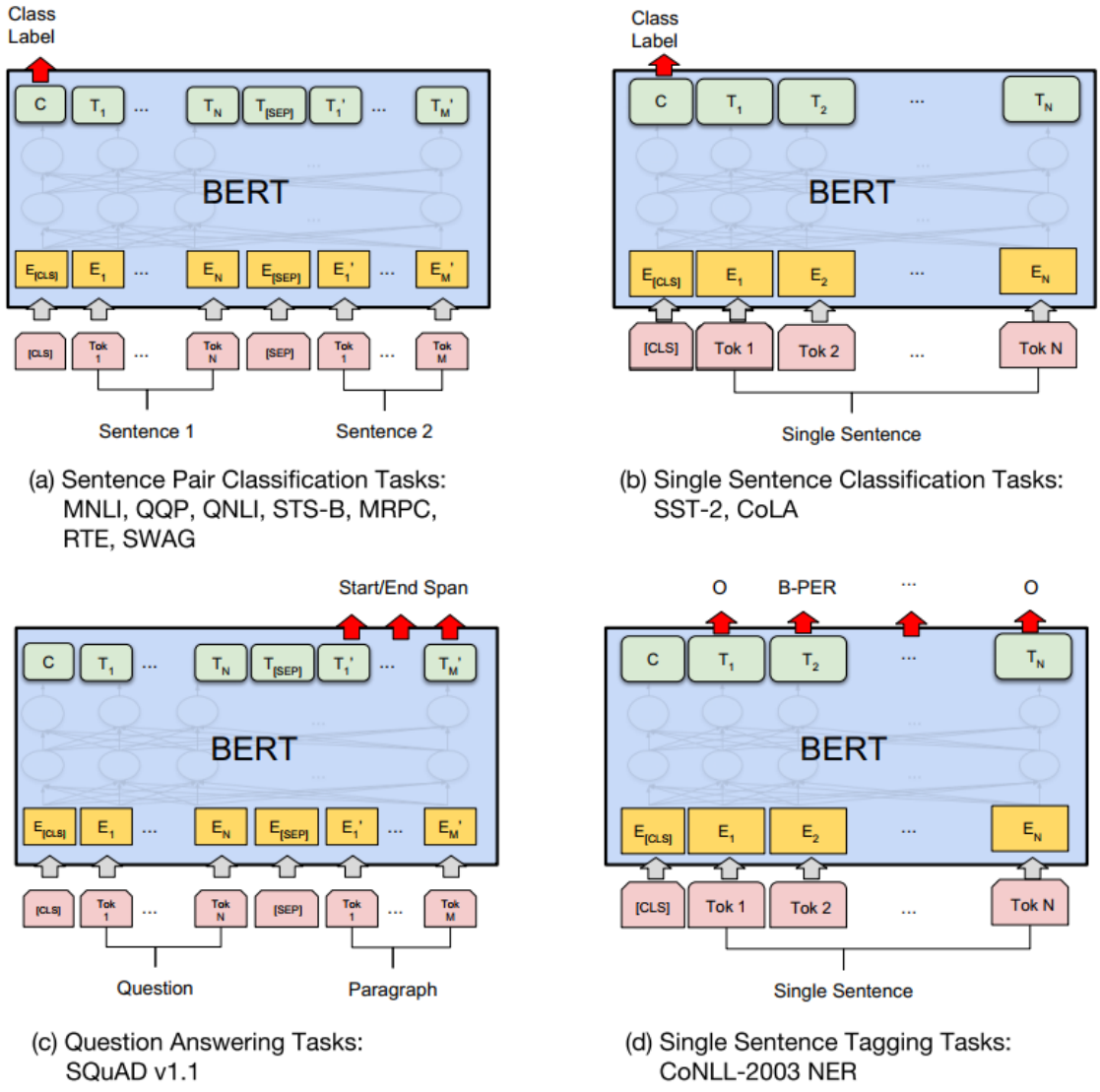

Figure 1. usage of BERT for different downstream tasks

3. Our Work

3.1. Description of the data

We use a part of film scripts as training data which has been manually labeled with the character and its relative emotion type and degrees. We are provided with another part of film scripts as testing data whose labels are removed by the competition host. We need to identify the emotions of each character involved in every dialogue and action description in the script scenes and give the emotion degrees.

Figure 2 illustrates a part of the training data: The table content contains the script of a movie. The character column contains the specified character, that is mentioned in the script. The last six columns are the labels, which are in the training data but missing in the test data. The task is to identify the given character's six emotions: love, happiness, surprise, anger, fear, and sorrow, and numerically rank them according to the script. A sentence has multiple characters, such as p2, d1 and x2, and for each character, the type and degree of emotion needs to be identified. In the sam-

id	id2	content	charact	love	joy	supprise	anger	fear	sorrow
1460_0016_A_	189	全家人站成一排，拍了一张全家照。The whole family stood in a line for a family photo.							
1460_0016_A_	190	d1挽着p2走出照相馆，门口停着一辆婚车。D1 walked out of the studio with P2 on his arm. There was	p2	0	0	0	0	0	0
1460_0016_A_	191	d1挽着p2走出照相馆，门口停着一辆婚车。D1 walked out of the studio with P2 on his arm. There was	d1	0	1	0	0	0	0
1460_0016_A_	192	一x2称赞：哇，靓车啊！A X2 praise: Wow,a beautiful	x2	0	2	3	0	0	0
1460_0016_A_	193	p2不小心摔倒在地，d1连忙扶起p2。P2 accidentally fell to the ground, D1 picked up P2	p2	0	0	2	0	0	0
1460_0016_A_	194	p2不小心摔倒在地，d1连忙扶起p2。P2 accidentally fell to the ground, D1 picked up P2	d1	0	0	2	0	0	0
1460_0016_A_	195	d1：没事吧？D1: Are you all right?	d1	3	0	0	0	0	0
1460_0016_A_	196	p2笑着：没事，没事。P2 smiled: Nothing, nothing	p2	2	3	0	0	0	0

Figure 2. Example of training data set.

degree	love	joy	surprise	anger	fear	sorrow
0'	28434	27262	27735	26397	27048	24898
1'	420	1645	1033	1612	1253	2259
2'	328	370	458	981	815	1594
3'	273	180	229	465	339	753
percentage of 0	0.965337	0.925485	0.941606	0.896181	0.918282	0.843886
percentage of 1	0.014259	0.055844	0.03507	0.054728	0.042539	0.076566
percentage of 2	0.011136	0.012561	0.015549	0.033305	0.027669	0.054027
percentage of 3	0.009268	0.006111	0.007775	0.015787	0.011509	0.025522

Figure 3. data distribution statistics.

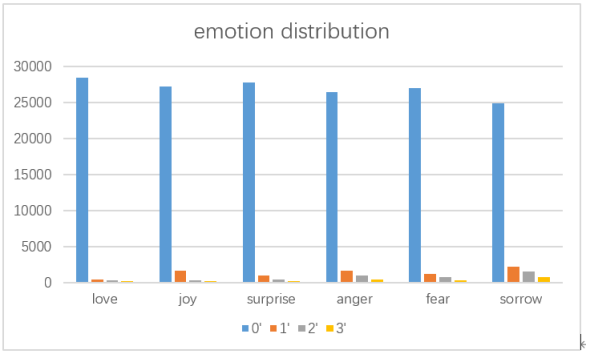

Figure 4. data distribution graph.

ple, there is one line: An x2 Praise: “Wow, beautiful car!”, which contains two emotions: ”joy” and ”surprise”, and they are in degree 2 and 3, respectively. The id of the data is composed of film id,scene id and sentence id, and separated by ”_”. The data is not strictly sorted by the id. The data with the same film id and scene id are more dependent to each other than that with different scene ids or film ids.

We shuffle and split the labeled data by the ratio of 8:2 and generate training set and validation set.We count and plot the emotion value distribution on the training set, as showed in Figure 3 and Figure 4. It is obvious that the data distribution is unbalanced. The emotion degree value 0 accounts for the vast majority and value 1 is the second majority, and the higher the emotional degree value is, the smaller proportion it takes.

3.2. Baseline Model

We implemented the baseline version using the simplest way by calling the existing python package simpletransformers MultiLableClassificationModel and got a score of 0.6814 on the test set and 0.6787 on the validation set. The details of the baseline model are given below(refer to Figure 5):

Since there are multiple emotion types (love, joy, surprise, anger, fear and sorrow) to be recognized, it is a multi-label classification problem. The category of each label has four values [0,1,2,3]. However, from the perspective of data distribution, category 2 and 3 account for a relatively small proportion. For simplicity, we only classify it into category 0 or 1, while category 2

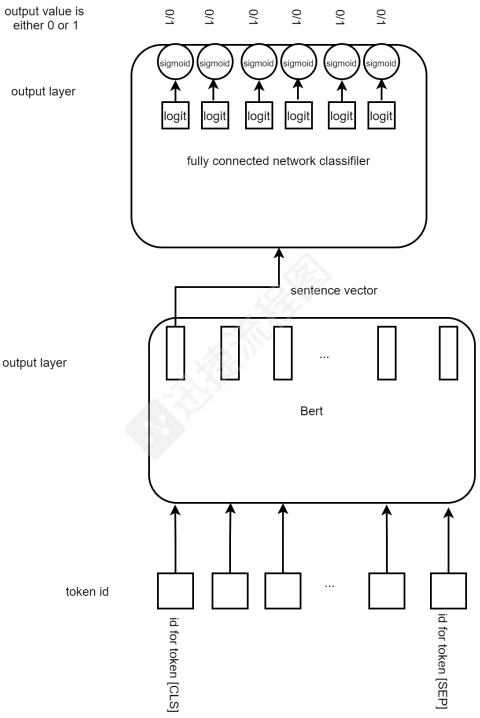

Figure 5. architecture of the baseline model.

and 3 are treated as category 1.

We combine character names and dialogues into one text, so that the emotion recognition of the character becomes a single text multi-label classification problem. Hence a multi-label binary classifier of simpletransformers package can be used.

For the sentence features, we adopt Bert-Base vector as the feature, which is currently popular and has a good performance.

We take a batch of text samples with a batch size of 8, calculate the maximum text length of this batch of data, convert each token of the text into token ID first, and transform a batch of text of different lengths into the id sequence of the same length, by padding those shorter sentence with 0.

Then input the token ID sequence into Bert model, and extract the vector from the first unit [CLS] in the last layer of the Bert model as the vector of the whole sentence.

Next send the sentence vector obtained into the fully-connected network for classification. Since there are six labels, the output logit is 6-dimension.

Different from the multi-classification, which does softmax with 6-dimension logit, while the multi-label classification task is to put the six bits of logit into 6 sigmoid activation functions respectively, output probability of label being 1.

Finally,convert output with probability greater than 0.5 to label 1, and convert output with probability less than or equal to 0.5 to 0.

The loss function adopts binary cross-entropy in training phase.

We choose the hyper-parameter max-epoch as 2, the default value for all other hyper-parameters.

3.3. The drawbacks of baseline model

In order to implement the task quickly, for the baseline version, we use the ready-made toolkit simpletransformers to build the model. we simplify the recognition of emotional value to a multi-label binary classification problem. In fact, there are four values of emotion: 0,1,2, and 3, which belong to multiple categories. However, simpletransformers doesn't support multi-label multi-classification.

Different from ordinary multiple category classification problem, each category is the intensity value of emotion, which is actually numeric value and comparable in size. If it is regarded as a general classification problem, using cross-entropy loss, then it cannot reflect the fact that the error between intensity value 1 and intensity value 3 is larger than the error between intensity value 1 and intensity value 2, so intuitively we think it makes more sense to turn to regression rather than categorization.

Since emotion recognition is aimed at the designated script character, the baseline method is simply to combine the characters of the script into the text of the script, so as to classify the text. But characters play a crucial role here, and if we combine them into text, we treat them like any other word in the text, which gives the model a chance

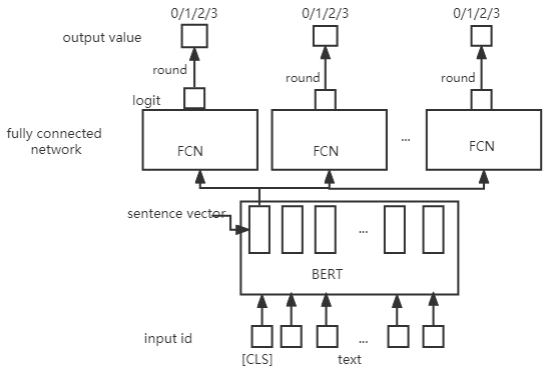

Figure 6. architecture of the improved model 1.

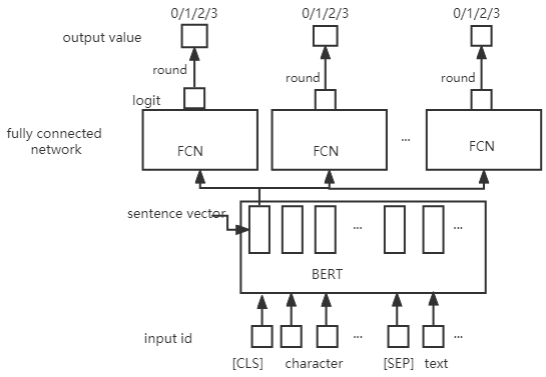

Figure 7. architecture of the improved model 2 and 3. The only difference is that the text of the improved model 3 includes not only current text but also the historic texts.

to overlook the importance of the script character.

Through data analysis, we know that the emotional value is not only related to the current text, but also related to historic text within the same scene. The same sentence in different contexts will show different emotions, while baseline model does not make use of the above information. Therefore, based on the above analysis, we gradually made improvements, thus gradually improving the scores of validation set and test set.

3.4. Improved Model 1

Figure 6 shows the architecture of the improved model 1. We modify the output parts and keep the other parts unchanged.Different from baseline model, We adopt multiple output blocks, one emotion corresponds to one output block, and there are two layer fully-connected network inside each output block. The output dimension of the last layer is 1, without adding any activation function, using MSE as the loss function. The resulting logit is a float number, and we round logit to get integers in the range of [0,1,2,3]. The emotional values of the six emotions correspond to the integral values of the six output blocks. Since there are multiple output blocks, the total loss of the model is the summation of the losses of all outputs. It's actually a multitasking approach to learning. Each task is to identify one of the emotions, but share sentence vectors. Fine-tuning technique is also adopted. We train BERT parameters together with the output blocks,but with different learning rates. We choose output blocks learning rate 1e-04 and BERT learning rate 1e-05.

3.4.1 Improved Model 2

On the basis of Improved Model 1, with reference of the way BERT model handles question-and-answer task, regarding script character as question and script text as paragraph(see Figure 1 c.Question Answering Tasks:SQuAD v1.1),we feed them to the BERT. However, different from question-and-answer task, which gets the answer's start and end position in the sequence of the token, we still take the hidden vector from the first position of the output layer as a vector representation of the script text and the script character pair. This vector representation is fed into each output block as a feature as shown in Figure 7. Since script character and script text implement cross attention with each other inside the Bert, the model attaches much more importance to the character than baseline model does.

Name	Validation dataset	Test dataset
Baseline Model	0.6787	0.6814
Improved Model (1)	0.6837	0.6816
Improved Model (2)	0.6860	0.6842
Improved Model (3)	**0.6907**	**0.6864**

Table 1. scores of different models on validation dataset and test dataset

3.4.2 Improved Model 3

On the basis of Improved Model 2, we introduce historical text. Since the text length of the model cannot be increased infinitely, we only quote previous two sentences for the time being. We simply merge the previous two sentences with the current sentence to get a longer text. In order to find and use the historical text, we pre-process the data. We extract all the script text ,remove duplicates of it and sort it by the order of script id,scene id and sentence id to generate a unique and sorted script text list. We give an new id "content_id" to the unique script text and associate the unique content_id to the every single data. During training phase or inference phase, we firstly find the content_id of the original training data or test data, and then get the three corresponding texts of content_id-2,content_id-1 and content_id from the script text list and merge them into one longer text in sequence. We use the same architecture of Improved Model 2 shown in Figure 7, and feed the longer text and script character into the model.In this way, we use historical information to predict the character emotion of the current text.

3.5. Results

Table 1 shows the performance results for different models. The formulation to calculate the score is discussed in section Performance measure metric. The baseline model gets the lowest score,improved model 1 gains the improvement of 0.005 on the validation dataset and slight improvement on test dataset. Improved Model 2 gains 0.0023 more on the validation dataset and 0.0026 more on the test dataset. Improved Model 3 gets the highest score on both datasets,0.6907 and 0.6864 which are 0.012 and 0.005 higher comparing to baseline model respectively. This shows that our improved methods are working.

4. Future work

Currently we feed the text into BERT directly,but BERT can't receive too long text. The max length of text of BERT is 512 tokens, but our max length is 300 tokens and longer length might result in out of GPU memory easily. In the future, we plan to try adding an attention layer between the fully connected network and BERT. In stead of merging the all sentences into one longer text,we extract every sentence's BERT vector and feed them to the attention layer. In this way, we can make use of more historic sentences to predict the emotion.

5. Conclusion

We studied the film script emotion recognition problem which is more challenging than previous sentiment analysis or multi-label emotion analysis, we proposed baseline model using an existing python package simpletransformers and in order to use the existing package,we simplify the problem. And we devised three improved methods which gain better performance. The first method supports multi-label multi-classification that the existing package simpletransformers doesn't support. We further propose that using regression (later converting the float output into integer output) instead of classification to classify numeric values. In the second method, we input the script character and script text separately to avoid the

model ignore the importance of script character. In the third method we make use of the historic sentences. Every method can improve the scores in both validation dataset and test dataset significantly.

References

[1] D. Demszky, D. Movshovitz-Attias, J. Ko, A. Cowen, G. Nemade, and S. Ravi. Goemotions: A dataset of fine-grained emotions.

[2] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert:pretraining of deep bidirectional transformers for language understanding. 2019.

[3] V. Tripathi, A. Joshi, and P. Bhattacharyya. Emotion analysis from text: A survey. 2016.

[4] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, and Łukasz Kaiser. Attention is all you need. 2017.

[5] W. Ying, R. Xiang, and Q. Lu. Improving multi-label emotion classification by integrating both general and domain knowledge.

[6] L. Zhang, S. Wang, and B. Liu. Deep learning for sentiment analysis: A survey.