DOI: 10.54254/2755-2721/38/20230525

Sentiment analysis based on BiLSTM with attention mechanism on Chinese comment with stickers

Yuchen Bao[1,4], Hongyi Huang[2] and Zizhou Meng[3]

[1] College of Artificial Intelligence, Jianghan University, Wuhan, 430056, China
[2] Malvern college chengdu, Chendu, 610401, China
[3] Beijing No.13 High School, Beijing, 100009, China

[4] yuchen_bao@stu.jhun.edu.cn

Abstract. As the Internet is progressively becoming larger and more intricate, more and more users of various social media choose to post their comments to express their opinions and thinking on those platforms. Analyzing the emotions contained in user comments holds great business value, helping to accurately perceive user consumption habits and improve user service levels. However, the use of emoticons and stickers in comments has increased dramatically in recent years, which brings new challenges to text sentiment analysis based on natural language processing. In this paper, in order to alleviate the above problems, we propose a method for analyzing the sentiment of Chinese comments based on the attention mechanism and BiLSTM. Specifically, we partitioned the original dataset from the Weibo platform according to the number and type of emoticons in the comments. By analyzing the actual data, the specific features of emojis that affect the performance of sentiment analysis are identified, and corresponding explanations are given. In addition, a hypothesis is proposed to quantify the impact of emoticons on model effectiveness. All the results demonstrate the effectiveness of our proposed method.

Keywords: sentiment analysis, BiLSTM, attention mechanism.

1. Introduction

Sentiment analysis (SA) is an important field of NLP and SA technology can classify the emotions of various types of texts. Serving as platforms for open expression, social networks can be vital sources of information for analyzing public sentiment. This is particularly relevant for significant public events, as they can provide crucial insights to inform decision-making [1]. In recent years, with the advancement of software technology and front-end development, people have increasingly engaged in the frequent usage of social networks. Moreover, within the comments they post, the utilization of emoticons such as stickers and emojis has become exceedingly prevalent. Relevant studies have also identified that the utilization of emoticons in communication often enhances the emotional intensity of text expression. On occasion, these symbols can synergize with text to convey deeper emotional nuances. Emoticons have been empirically shown to enrich the spectrum of emotional expression, making them an equally vital feature of textual communication [2][3].

Broadly speaking, emoticons refer to symbols visualized as cartoon-like expressions by users. However, fundamentally, emoticons are primarily categorized into several types: emojis, stickers, and memes. Emojis are essentially determined by unique codes similar to Chinese characters, whose number of characters available for users can significantly differ depending on the system and version, though their display forms remain relatively consistent. Stickers are essentially images, presented to users as cartoon expressions by websites or apps through the addition of special markers within the text. Common marker methods include " "(WeChat, QQ) and '[]' (Weibo, Xiaohongshu, Baidu Tieba). In various operating systems or different versions of operating systems, as long as they belong to the same company's products, there won't be differences. However, the distinctions between different products are significant, to the extent that even the naming conventions differ. Memes, also image-based, lack textual representation and exhibit identical display forms regardless of dissemination platform. However, they tend to be more associated with image processing and carry more intricate information.

In past studies, researchers always used traditional machine learning algorithms or deep learning models to analyze texts with emoticons, especially emoji. While traditional machine learning models solve this problem with a carefully labeled sentiment lexicon, deep learning focuses on model changes to get more information or by training word embedding that is more tailored to the specific application context for better results [4][5][6]. Due to the rapid iteration of internet language, traditional manual methods for constructing sentiment lexicons have become prohibitively costly and time-consuming. As a result, semi-supervised and unsupervised approaches for sentiment lexicon construction have emerged [7], which decreases the time cost and gets good accuracy. The AEC-BiLSTM model proposed by Hang et al. in 2021 achieves an accuracy of 0.96 on a binary classification dataset of IMDB hotel reviews [8]. Further, there are commonly two ways to boost the effect of models. The First way is optimizing the word embedding to ensure models can learn higher-quality information. The CEmo-LSTM model proposed by Liu et al. achieves an accuracy of 0.95 when processing Chinese text containing emoticons [4]. The second way is to find novel models or a combination of models or add some new mechanisms to improve an existing model. Researchers also added an emoji attention mechanism on BiLSTM to analyze the sentiment more deeply, and the accuracy of the model can reach 0.87 in a dataset labeled three types of emotion [9].

However, there is a significant overlap between stickers and emojis in terms of functionality. Many companies started to develop their own sticker so more customers could use their products in any operating system. Even the naming and display of many of the stickers aim to be similar to emoji. The trend of stickers replacing emojis as the internet grows is evident in Asia, and the trend is starting to appear in the West as well [10]. Hence, investigating the impact of stickers on text sentiment analysis is of paramount importance. Nevertheless, prior research has predominantly focused on analyzing the influence of emojis on text sentiment analysis or on devising novel models to better handle text containing emojis, often neglecting to allocate significant attention to stickers. In this paper, we will build a BiLSTM model with an attention mechanism, then divide the dataset according to the different features of the STICKERS, and then analyze the effect of these stickers on the model's effectiveness.

2. Method

2.1. Structure of BiLSTM

The Long Short-Term Memory (LSTM) network possesses robust capabilities in addressing the challenge of long-range dependencies in text, while also mitigating the issue of vanishing gradients [11]. It achieves this by selectively retaining and transmitting information from the text to subsequent layers. LSTM models are frequently employed for prediction and natural language processing (NLP) tasks, such as stock price forecasting [12][13], text sentiment classification, and information extraction [14][15].

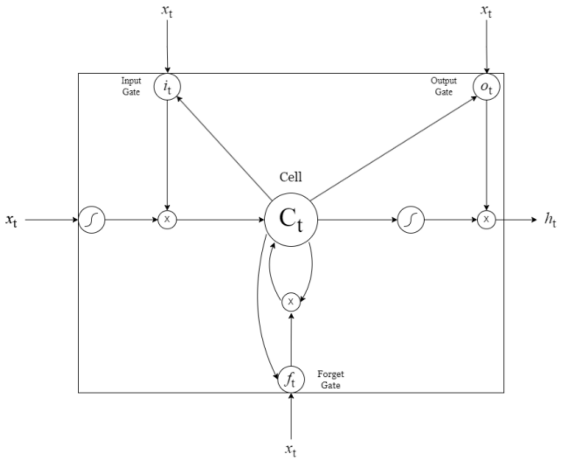


Figure 1. The structure of LSTM unit [11].

As shown in Figure 1, each computational unit of an LSTM model employs forget gates, memory gates, and output gates to ensure that features from distant positions in the text can be propagated to later positions. This mechanism enables the model to capture and utilize long-range dependencies effectively. The whole calculation process of LSTM can be formulated as:

$$h(t) = f(U_x(t) + W_h(t-1)) \tag{1}$$

$$y(t) = g(V_h(t)) \tag{2}$$

Similar to RNN, in LSTM, x represents the input, y is the output, and h signifies the output of the hidden layer. U and W denote the weights for the input x and the previous hidden layer's output, respectively. f(x) and g(x) represent the sigmoid activation function and softmax activation function, respectively. LSTM's computational unit introduces additional components like the cell state, forget gate, input gate, and output gate to retain and forget features from the text, enhancing its capability to capture and utilize contextual information, as:

$$i_t = \sigma(W_{X_i}x_t + W_{h_i}h_{t-1} + W_{c_i}c_{t-1} + b_i) \tag{3}$$

$$f_t = \sigma(W_{x_f}x_t + W_{h_f}h_{t-1} + W_{c_f}c_{t-1} + b_f \tag{4}$$

$$c_t = f_t c_{t-1} + i_t\, tanh(W_{x_c}X_t + W_{h_c}h_{t-1} + b_c) \tag{5}$$

$$o_t = \sigma(W_x o_x + W_{h_o}h_{t-1} + W_{h_o}h_{t-1} + W_{c_o}C_t + b_o) \tag{6}$$

$$h_t = o_t\, tanh(c_t) \tag{7}$$

where i, f, c, and o respectively represent the input gate, forget gate, cell state, and output gate. They each process vectors with dimensions matching that of the input x. W and b stand for weight parameters. h_t represents the hidden layer output at each time step.

$$O_t = \sigma(V o_{t_f} + V o_{t_b} + C_o) \tag{8}$$

In bidirectional LSTM, each output of the model is determined by the combined outputs from both directions. O_t represents the final output of the model at each time step. V and C are weight parameters, and o_{t_f} and o_{t_b} represent the outputs propagated forward and backward, respectively.

2.2. Attention mechanism

In order for the model to notice useful information in the text, after the features are extracted by the BiLSTM layer, the model also decides which time step is more important according to the attention mechanism and then integrates these features into a sentiment feature vector for the whole sentence. s denotes the overall sentiment feature of the entire sentence:

$$s = \sum_{t=1}^{T} a_i \tag{9}$$

Where integer T represents the maximum number of time steps contained within the sequence, $\mathrm{t} \in [1, \mathrm{T}]$. $\mathrm{a_i}$ represents the features noted at each time step after weighting:

$$\mathrm{a_t} = \mathrm{softmax}(\mathrm{score}(\mathrm{h_t}))\mathrm{h_t} \tag{10}$$

The softmax function, as defined, allocates weights to all elements based on the values of $\mathrm{z_i}$, $\mathrm{z_i} \in \mathrm{Z}$:

$$\mathrm{softmax}(\mathrm{z_i}) = \frac{\mathrm{e^{z_i}}}{\sum_{\mathrm{j=1}}^{\mathrm{n}} \mathrm{e^{z_j}}} \tag{11}$$

The $\mathrm{score}(\mathrm{h_t})$ maps the hidden layer features at each time step to the [0,1] interval and then multiplies them with the feature extraction matrix w, resulting in a sentiment score associated with the textual information for that specific time step.

$$\mathrm{score}(\mathrm{h_t}) = \tanh(\mathrm{h_t}) \times \mathrm{w} \tag{12}$$

where w is a learnable parameter that becomes more suitable for extracting emotional features of the text during the training process.

2.3. Model training

The loss function for training is the cross entropy loss function, which is defined as following equation:

$$\mathrm{loss} = -\sum_{\mathrm{m=1}}^{\mathrm{M}} \sum_{\mathrm{k=1}}^{\mathrm{K}} \mathrm{p}(\mathrm{x_k}) \times \log_2(\mathrm{Q}(\mathrm{x_k})) \tag{13}$$

Where K represents the total number of emotion categories, $\mathrm{p}(\mathrm{x_k})$ represents the probability vector of labeled emotion as $\mathrm{x_k}$, $\mathrm{Q}(\mathrm{x_k})$ represents the probability vector of predicted emotion as $\mathrm{x_k}$, and m denotes the number of texts included in a single training instance.

2.4. Structure of BiLSTM-attention model

The network structure of BiLSTM-Attention model is shown in Figure 2. Here, CLt and hLt represent the cell state and hidden layer output during the forward pass of the model, and Crt and hrt represent the cell state and hidden layer output during the reverse pass. The combined hidden layer output ht symbolizes the result of concatenating the forward and backward hidden layer outputs. These outputs undergo computations via the Attention mechanism to derive the overall sentiment feature of the entire text. Subsequently, they enter a fully connected layer for feature analysis, ultimately leading to the classification outcome.

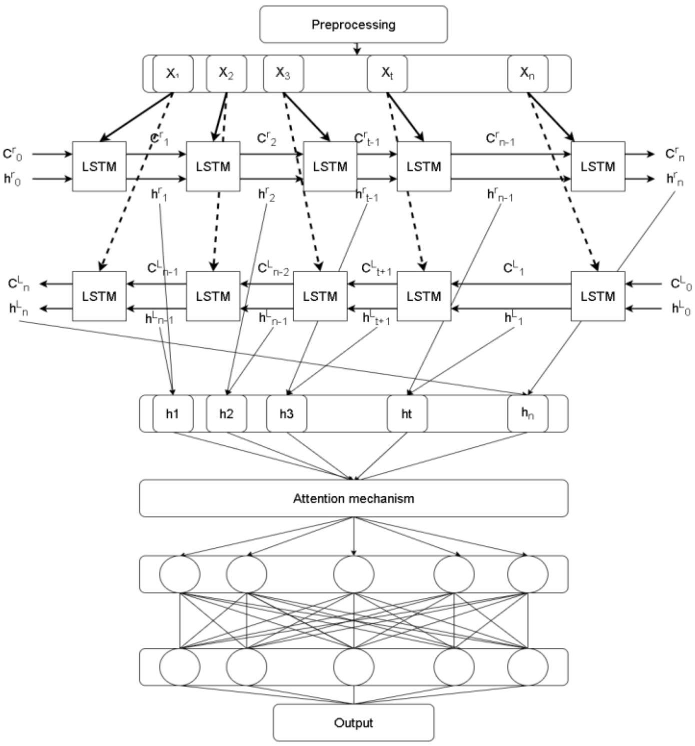


Figure 2. The network structure of BiLSTM-attention model.

2.5. Data sources and processing

In the whole experiment, the dataset consists of n real Weibo comments with stickers. In the overall dataset, the longest sample consists of 230 characters, while the shortest contains 3 characters. We counted the number and type of stickers included in different comments in the dataset, and their distribution is shown in Figure 3. The samples encompass a maximum of 65 stickers, with a minimum of 1 sticker. The training data comprises 55000 sampled examples from the overall dataset, distributed in proportion to 50% for both positive and negative sentiments. Among these, 50000 instances are employed as the training set, while 5000 instances serve as the validation set. The remaining instances constituting 64988 of the data, are allocated to the test set, with positive sentiments comprising 51% and negative sentiments comprising 49%. The test set samples encompass a maximum of 47 stickers and a minimum of 1 sticker. All those comes from real Weibo comments. The test set is divided according to the quantity and types of emoticons present, and is concurrently employed to evaluate the performance of the 40 generations of the model.

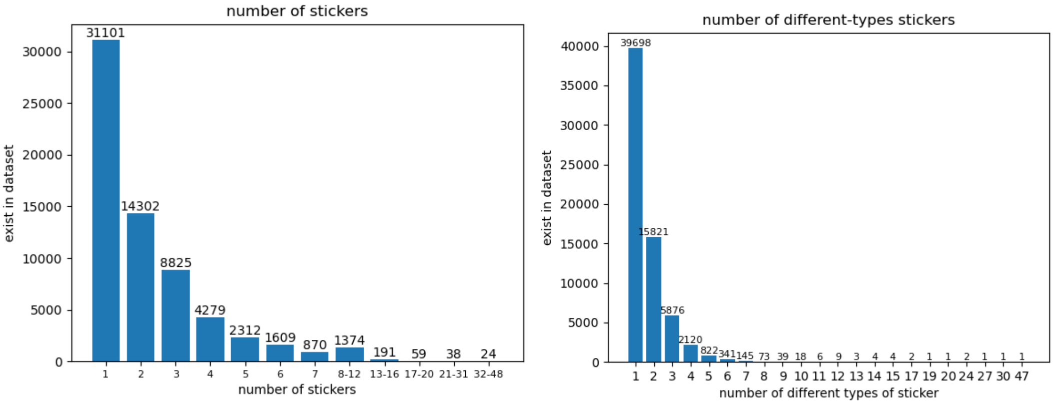


Figure 3. The distribution of sticker number and types.

3. Experiment

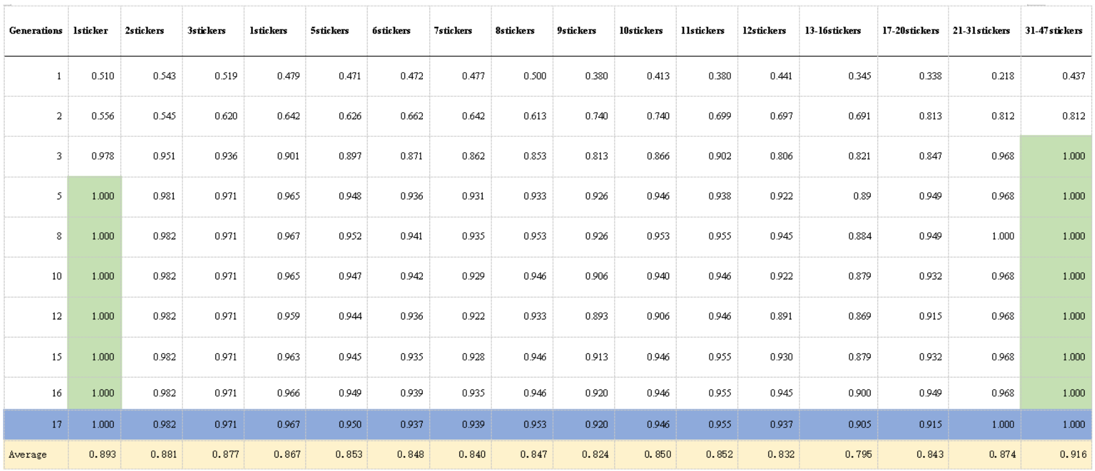
3.1. Partition of the test dataset

Since the frequency of occurrence of texts containing different stickers in the test dataset which contains 64988 texts varies greatly, in order to avoid the chance of experiments caused by too little data, in for those sets that appear a lot of emoticons, but the sample capacity is very small to do the merging process, and finally divided the dataset into 12 datasets, respectively, the set containing 1,2,3,4,5,6,7,8-12,13-16,17-20,21-31,32-47 stickers. Another test dataset was also constructed from the test dataset which contains 64988 texts by the types of expressions, which are the text datasets containing 1,2,3,4,5,6,7,8,9,10,11-15,16-47 expressions.

3.2. Result analysis

3.2.1. Analysis for different numbers of stickers

Different generations of the Attention-enhanced BiLSTM models are applied to test each dataset, and upon conducting the tests, the models yield the following results as shown in Figure 4. Through the cross-generational comparison of data and the assessment of the average values for each generational model across the same quantity of stickers, the accuracy of the model's judgments exhibits a pattern of initial increase followed by subsequent decline. This observation underscores that the quantity of sticker symbols indeed yields a discernible adverse impact on the accuracy of the model's judgments, irrespective of whether one considers the individual generational models or the overall aggregate performance.



Generations	1sticker	2stickers	3stickers	1stickers	5stickers	6stickers	7stickers	8stickers	9stickers	10stickers	11stickers	12stickers	13-16stickers	17-20stickers	21-31stickers	31-47 stickers
1	0.510	0.543	0.519	0.479	0.471	0.472	0.477	0.500	0.380	0.413	0.380	0.441	0.345	0.338	0.218	0.437
2	0.556	0.545	0.620	0.642	0.626	0.662	0.642	0.613	0.740	0.740	0.699	0.697	0.691	0.813	0.812	0.812
3	0.978	0.951	0.936	0.901	0.897	0.871	0.862	0.853	0.813	0.866	0.902	0.806	0.821	0.847	0.968	1.000
5	1.000	0.981	0.971	0.965	0.948	0.936	0.931	0.933	0.926	0.946	0.938	0.922	0.89	0.949	0.968	1.000
8	1.000	0.982	0.971	0.967	0.952	0.941	0.935	0.953	0.926	0.953	0.955	0.945	0.884	0.949	1.000	1.000
10	1.000	0.982	0.971	0.965	0.947	0.942	0.929	0.946	0.906	0.940	0.946	0.922	0.879	0.932	0.968	1.000
12	1.000	0.982	0.971	0.959	0.944	0.936	0.922	0.933	0.893	0.906	0.946	0.891	0.869	0.915	0.968	1.000
15	1.000	0.982	0.971	0.963	0.945	0.935	0.928	0.946	0.913	0.946	0.955	0.930	0.879	0.932	0.968	1.000
16	1.000	0.982	0.971	0.966	0.949	0.939	0.935	0.946	0.920	0.946	0.955	0.945	0.900	0.949	0.968	1.000
17	1.000	0.982	0.971	0.967	0.950	0.937	0.939	0.953	0.920	0.946	0.955	0.937	0.905	0.915	1.000	1.000
Average	0.893	0.881	0.877	0.867	0.853	0.848	0.840	0.847	0.824	0.850	0.852	0.832	0.795	0.843	0.874	0.916

The accuracy of different generations of model on datasets of different number of stickers

[1] **The generation of models not means epochs.In training process,the model which has both relatively lower loss and higher accuracy can be saved as the next generation of models.**

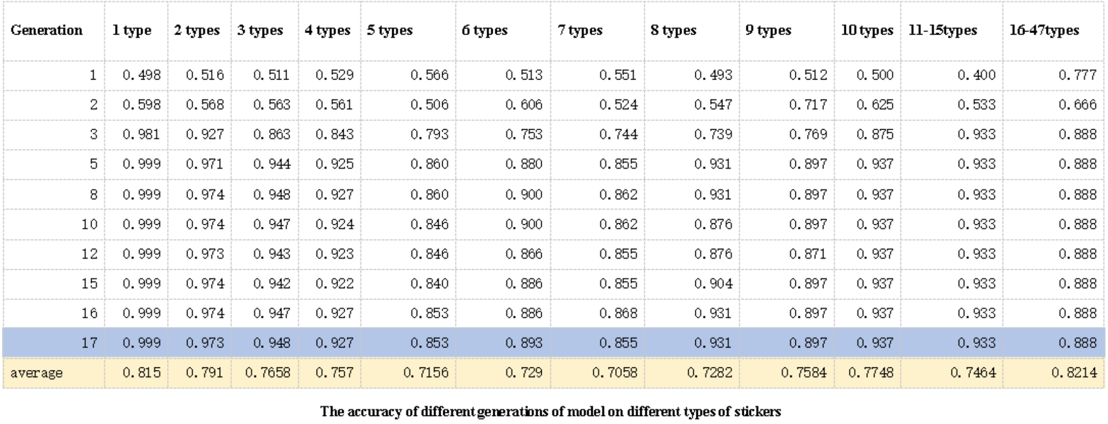
Figure 4. Model performance for comments with different numbers of stickers.

When scrutinizing the vertical comparison of models from different generations utilizing the same sticker test dataset, it becomes apparent that, for test sets containing a lower count of stickers, the optimal number of training epochs required for the model to attain its optimal performance follows a trajectory characterized by initial augmentation and subsequent reduction. This trend underscores the reality that, concerning the training of the model, an augmented abundance of sticker symbols often signifies the presence of more intricate informational nuances. As a result, the model necessitates an escalated number of iterative processes to adeptly optimize parameters for effectively handling these heightened complexities.

However, it is worth noting that both in horizontal and vertical comparisons, the model's performance tends to converge towards test sets with a lower count of stickers (17-20 stickers, 21-31 stickers, 31-47 stickers). For instance, the performance of models on test sets with 21-30 stickers, 31-47 stickers, and even on sets with just 1 sticker, exhibits remarkable similarities. Each of these models ultimately achieves nearly perfect accuracy rates. Additionally, it's notable that the convergence patterns observed in the 31-47 stickers test set closely resemble those witnessed in the scenario involving only 1 sticker. This observation underscores intriguing insights into the behavior of the model in handling varying levels of sticker complexity within the dataset, which carries relevance to practical applications within the realm of computer science and engineering.

3.2.2. Analysis for different types of stickers

Through an analysis of the dataset, we have found that within natural microblogging datasets, samples with a notably high number of sticker symbols often tend to employ the same stickers. This observation indicates that the quantity of sticker symbols can indeed serve as an indicator of data information complexity. However, it is not the sole influencing factor. The specific variety of stickers present within a sentence might also impact the effectiveness of sentence-level models. Based on the dataset detailing the occurrence frequencies of different sticker types within sentences, the model has yielded the following data in Figure 5.



Generation	1 type	2 types	3 types	4 types	5 types	6 types	7 types	8 types	9 types	10 types	11-15types	16-47types
1	0.498	0.516	0.511	0.529	0.566	0.513	0.551	0.493	0.512	0.500	0.400	0.777
2	0.598	0.568	0.563	0.561	0.506	0.606	0.524	0.547	0.717	0.625	0.533	0.666
3	0.981	0.927	0.863	0.843	0.793	0.753	0.744	0.739	0.769	0.875	0.933	0.888
5	0.999	0.971	0.944	0.925	0.860	0.880	0.855	0.931	0.897	0.937	0.933	0.888
8	0.999	0.974	0.948	0.927	0.860	0.900	0.862	0.931	0.897	0.937	0.933	0.888
10	0.999	0.974	0.947	0.924	0.846	0.900	0.862	0.876	0.897	0.937	0.933	0.888
12	0.999	0.973	0.943	0.923	0.846	0.866	0.855	0.876	0.871	0.937	0.933	0.888
15	0.999	0.974	0.942	0.922	0.840	0.886	0.855	0.904	0.897	0.937	0.933	0.888
16	0.999	0.974	0.947	0.927	0.853	0.886	0.868	0.931	0.897	0.937	0.933	0.888
17	0.999	0.973	0.948	0.927	0.853	0.893	0.855	0.931	0.897	0.937	0.933	0.888
average	0.815	0.791	0.7658	0.757	0.7156	0.729	0.7058	0.7282	0.7584	0.7748	0.7464	0.8214

The accuracy of different generations of model on different types of stickers

Figure 5. Model performance of various generation models on different types of stickers.

When the data consists of only a single type of sticker (which might appear 2 times, 3 times, or even more within a sentence), the model's final accuracy is also remarkably close to 1. However, there still exists a minimal disparity compared to the accuracy on test sets featuring only one sticker. This observation further substantiates the conjecture that the quantity of stickers does indeed influence the model's efficacy to a certain extent.

In the horizontal comparison, the model's accuracy continues to exhibit the trend of an initial decrease followed by a period of stability and ultimately an ascent. Nonetheless, the overall accuracy of the model experiences a noticeable decline when compared to the division based on sticker quantity. While the model's accuracy remains consistently above 0.9 in the division by sticker quantity, the lowest accuracy within the division by sticker type is only 0.853. This underscores that the category of stickers within the model is a pivotal factor influencing accuracy, and its impact on model performance is more pronounced.

Although the model's accuracy does show some recovery on the test sets with a greater variety of sticker types, there exists an inherent margin of error. This is particularly evident in the 11-15 types test set with only 26 instances, and the 16-47 types test set with a mere 9 instances. Test sets with such a limited amount of data can yield results with a significant level of randomness. Nonetheless, even in light of this, the model's accuracy on these two test sets remains notably lower than the final model accuracy achieved through division based on sticker quantity.

4. Discussion

Both the number of stickers in the text and the type of sticker affect the training and final results of the model. The higher the number of identical stickers, the higher the accuracy of the model. The higher the number of different kinds of stickers, the lower the accuracy of the model.

Based on the above experiments and data, it can be assumed that a sticker is equivalent to a special information representation character, when there are more types of stickers and the distribution of the number of each sticker is more uniform, the higher the complexity of the sticker features, the worse the model's effect; while the fewer the types of stickers and the more the number of each sticker is and the distribution is polarized, the lower the complexity of the expression features, the better the model's effect. We can use the information entropy to represent the complexity of emoticon or sticker features:

$$H(X) = \sum_{k=1}^{n} p(x_k) \times \log_2 p(x_k) \tag{14}$$

Where X is a random variable of all emojis in a text, H(X) represents the complexity of the emoji information in the text where X random variable is located, k represents the number of emojis, and p(x) represents the probability of emoji x appearing in the text, which reflects the number of times the emoji appear in the text from the side. This formula can very intuitively reflect the impact of the number and type of emoji on the sentiment classification results. The higher the complexity, the worse the result, and the lower the complexity the better the result.

5. Conclusion

With the development of the Internet, the sticker has been more and more widely used in social media platforms, as an important symbol for expressing emotions, sticker also contains very complex emotional information. In the sentiment analysis of these real texts, both the number of stickers and the types of stickers have an impact on the results of sentiment analysis, with the latter having a more significant impact. But overall this effect is attributed to the probability of each sticker appearing relative to all stickers in the text. In future work, this impact can be measured using larger datasets in order to find more precise and appropriate ways to characterize and even quantify this impact.

References

[1] Alam K N, Khan M S, Dhruba A R, et al. Deep learning-based sentiment analysis of COVID-19 vaccination responses from Twitter data[J]. Computational and Mathematical Methods in Medicine, 2021, 2021.

[2] Shiha M, Ayvaz S. The effects of emoji in sentiment analysis[J]. Int. J. Comput. Electr. Eng. (IJCEE.), 2017, 9(1): 360-369.

[3] Li M, Ch'ng E, Chong A Y L, et al. Multi-class Twitter sentiment classification with emojis[J]. Industrial Management & Data Systems, 2018, 118(9): 1804-1820.

[4] Liu C, Fang F, Lin X, et al. Improving sentiment analysis accuracy with emoji embedding[J]. Journal of Safety Science and Resilience, 2021, 2(4): 246-252.

[5] Peng H, Cambria E, Hussain A. A review of sentiment analysis research in Chinese language[J]. Cognitive Computation, 2017, 9: 423-435.

[6] Wu J, Lu K, Su S, et al. Chinese micro-blog sentiment analysis based on multiple sentiment dictionaries and semantic rule sets[J]. IEEE Access, 2019, 7: 183924-183939.

[7] Zhang L, Wang S, Liu B. Deep learning for sentiment analysis: A survey[J]. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 2018, 8(4): e1253.

[8] Huang F, Li X, Yuan C, et al. Attention-emotion-enhanced convolutional LSTM for sentiment analysis[J]. IEEE transactions on neural networks and learning systems, 2021, 33(9): 4332-4345.

[9] Lou Y, Zhang Y, Li F, et al. Emoji-based sentiment analysis using attention networks[J]. ACM Transactions on asian and low-resource language information processing (TALLIP), 2020, 19(5): 1-13.

[10] Konrad A, Herring S C, Choi D. Sticker and emoji use in Facebook Messenger: Implications for graphicon change[J]. Journal of Computer-Mediated Communication, 2020, 25(3): 217-235.

[11] Huang Z, Xu W, Yu K. Bidirectional LSTM-CRF models for sequence tagging. arXiv 2015[J]. arXiv preprint arXiv:1508.01991, 2015.

[12] Lu W, Li J, Wang J, et al. A CNN-BiLSTM-AM method for stock price prediction[J]. Neural Computing and Applications, 2021, 33: 4741-4753.

[13] Jin Z, Yang Y, Liu Y. Stock closing price prediction based on sentiment analysis and LSTM[J]. Neural Computing and Applications, 2020, 32: 9713-9729.

[14] Rehman A U, Malik A K, Raza B, et al. A hybrid CNN-LSTM model for improving accuracy of movie reviews sentiment analysis[J]. Multimedia Tools and Applications, 2019, 78: 26597-26613.

[15] Hsieh Y H, Zeng X P. Sentiment analysis: An ERNIE-BiLSTM approach to bullet screen comments[J]. Sensors, 2022, 22(14): 5223.