COMP90049 Project 2 Report: Which emoji is missing?

Anonymous

1 Introduction

The aim of this project is to analysis effectiveness of Naive Bayes algorithm on predicting the missing emoji in a sentence.

2 Data Set

The data set used to build training model and test in this report is from Twitter. There are two data sets split by different statistical methods. The Top 10 data set is based on Mutual Information and Chi-Square, while another one, Most 100, is using the greatest frequency in the training collection. The data sets contain 3 main parts of the attributes, which are identity of the sentence, lists of frequency of the words and emoji used. There are 10 types of emoji in each of data set, each type only corresponds to a sentence. This means that the sentences containing more than one emoji have been filtered.

The data in the each raw data set has split into 3 sets, which are train, development and test. However, in this report, only train and development data will be considered to be evaluated.

3 Evaluation Metrics

The following terms will be used in determining effectiveness:

- True Positive (TP), Positive instances classified as Positive
- False Negative (FN), Positive instances classified as Negative
- False Positive (FP), Negative instances classified as Positive
- True Negative (TN), Negative instances classified as Negative

For calculating the evaluation metrics, following equation will be used:

- Recall: proportion of positive tokens with correct prediction, calculated from $\frac{TN}{TP+FN}$
- Precision: proportion of correct result among all positive predictions, calculated from $\frac{TP}{TP+FP}$

4 Original Model

In this project, the tool of constructing model and train data set is Weka with Graphic User Interface (GUI). Because there are two data set built by different collection methods, the lists of frequency of word are different. Specifically, Top 10 collects 98 attributes while Most 100 has 100 attributes on building train models and testing. In the later of this report, Top 10 will be labeled as T and Most 100 will be the M.

When training the data set in Machine Learning, the classification method used is Naive Bayes. This classifier predicts the probability distribution of those chosen attributes where all of the attributes are mutually independent (Leung, 2007). The specific calculation of the probability is shown below:

$$p(c_i|x) = p(c_i) \times \prod_{k=1}^{n} p(x_k|c_i)$$

where c presents the class and x is the instance of each attribute. In each of collections, train set is used to train data and development set is for testing the performance. The Table 1 shows the overall results of two collections using Naive Bayes classifier. As can be seen from the table, the overall performance of M is better than the T, except the precision in T, which is higher than M. This may because the collection method in T predicts more valuable attributes. In the following paragraphs, the modification will mainly based on method M.

	T	M
TP Rate	26.2%	30.4%
FP Rate	9.7%	8.8%
Precision	44.1%	33.0%
Recall	26.2%	30.4%

Table 1: The overall performance of each collection method

The Table 2 shows the performance of individual emoji using original method M. Because they will be changed as the algorithm changes, the analysis will be based on the emoji which increases the performance.

	Precision (%)	Recall (%)
Clap	24.5	53.4
Cry	36.1	29.4
Disappoint	19.1	14.1
Explode	45.9	26.1
FacePalm	23.2	14.8
Hands	62.9	37.8
Neutral	25.0	23.5
Shrug	20.8	18.4
Think	29.8	38.5
Upside	27.9	27.9

Table 2: Performance of each emoji

5 Class Analysis

The number of count of each type emoji is illustrated in Table 3. It is obvious that 'Cry' and 'Upside' are the most popular emoji, whereas 'Disappoint' and 'FacePalm' are rarely used. Therefore, by reducing the cost weight of rarely used words, the performance would be improved.

	Count
Clap	3786
Cry	4820
Disappoint	1398
Explode	4118
FacePalm	1517
Hands	3744
Neutral	4580
Shrug	3756
Think	4400
Upside	5024

Table 3: The Number of Count of Each Emoji

In order to compute the re-weighted performance, cost sensitive classifier is used and the cost of disappoint is set to 0.3 and the cost of upside is set to 1.5 while others are 1. The result is shown in Table 4. It is clearly that the precision of 'Disappoint' and the recall of 'Upside' are improved significantly, though the recall and precision is dropped respectively. Meanwhile, the overall performance is increased slightly.

	Precision (%)	Recall (%)
Disappoint	32.7	11.5
Upside	25.3	35.5
Overall	33.6	30.5

Table 4: The optimized performance of variant form modification

6 Attribute Analysis

Attributes are the key section to determine the probability of emoji, analysis of important attributes can find a way improving the performance. In the following paragraphs, the details of the attributes that significantly affect the performance will be analyzed.

6.1 Repeated Words

The words may appear in attributes several times because of the different form, the two circumstances below list the common cases of the variations.

6.1.1 Different Tenses

The most common case is different tenses. For example, the variations of 'be' are shown in the collection T 4 times, which are 'been', 'is', 'are' and 'be' itself.

6.1.2 First-person pronouns

There are many variations of the first-person pronouns. For instance, the original model used the frequencies of 'i', 'im', 'ive' and 'ill' as attributes, it may cause problem of defects in classification. This is because, firstly, the prediction of Naive Bayes algorithm is based on the probability of attributes, but many of those first-person pronouns are not related to determining the type of emoji. That is to say, over-fitting of instances may be occurred if the first-person pronouns appears many times. Secondly, for example, 'i' and 'am' are related where the latter one must follow 'i'. However, there is no attributes showing the relationship of those. As the result, 'im' and 'am' will count separately and cause increasing inaccuracy.

6.1.3 Modification

Based on the problems mentioned in Section 6.1.1 and Section 6.1.2, all short forms of 'i' are

merged into 'i' by using adding up the frequencies of those words and for testing purposes, only tenses of 'be' are merged into 'be'. This can be done by using 'collections' library from Python to count the frequencies. As the result, the optimized results are shown in Table 5. As can be seen from the table, only half of them are improved performance. This is because if those modified words are not shown in a sentence, over-fitting may be occurred.

	Precision (%)	Recall (%)
Clap	23.8	51.3
Cry	37.2	31.3
Disappoint	19.1	14.5
Explode	42.8	23.3
FacePalm	24.0	15.0
Hands	62.5	37.4
Neutral	25.7	24.4
Shrug	20.7	18.6
Think	29.7	38.2
Upside	27.0	27.1

Table 5: The optimized performance of variant form modification

6.2 High Related Words

The cases below are high related to a specific emoji, that is, when those conditions are met, it is very likely to show a certain emoji.

6.2.1 Hyperlinks

Hyperlinks are used frequently in tweeters, there are some hyperlinks in attributes of the collection of T, such as 'httpstcobievlqvpm', 'httpstcoirudvwljda' and 'httpstcorwvydruvh', where the symbols of the hyperlinks are removed during the collection. When those hyperlinks are included in a sentence, it is very likely to show a certain emoji. For an instance, there are 100 tweeters which contains a hyperlink 'httpstcobievlqvpm' presenting 'Disappoint' emoji, while no any other emoji is shown. This may because an event related to the hyperlink makes people disappoint. Thus, considering the results of those hyperlinks will improve the performance largely.

6.2.2 Emoji Words

For the sentences that contain a specific emoji word, the probability of having the same emoji as the word is high. For an example, the tweeter id 3205 which is 'RT @admiringlegends: Im crying pls dont touch me bish' containing 'Cry' emoji word. Although the word is in the different tense, the emoji it used is 'Cry'.

6.2.3 Modification

In this modification, only the variants of 'cry' will be considered and one hyperlink related to the 'Cry' emoji will be added into the attributes. In order to compute the variants of the word 'cry', the Stanford CoreNLP Natural Language Processing Toolkit (Manning et al., 2014) will be used and the results are shown in Table 6. It is obvious that the performance remains the same, this may because the sentence which contains the variants of 'cry' or the hyperlink have already identified as 'Cry' emoji. Another reason may be the number of correcting is too small comparing with 12164 data to show the changes.

However, after deleting a hyperlink attribute 'httpstcobievlqvpm' in original method T, the precision of 'Disappoint' emoji has dropped slightly from 44.1% to 43.9%, which proves the assumption in Section 6.2.1 will affect the performance of its related specific emoji.

	Precision (%)	Recall (%)
Cry	36.1	29.4

Table 6: The optimized performance of variant form modification

7 Conclusions

In this report, a Machine Learning model is built using classifier Naive Bayes algorithm to identify which emoji used in a sentence. The original model is based on 2 collections and the optimizing method is based on modifying Most 100 method. Each optimization has separately discussed in the report and each change only improves the performance of some of emoji types, whereas the performance of other types either drops down or remains the same.

References

K Ming Leung. 2007. Naive bayesian classifier. *Polytechnic University Department of Computer Science/Finance and Risk Engineering.*

Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The stanford corenlp natural language processing toolkit. In *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*, pages 55–60.