COMP90049 Knowledge Technologies

Project 2: Which emoji is missing?

1. Introduction

In the present society, Twitter[1], as a typical representative of social media, is widely adopted around the world allowing users expressing their opinions in a limit of 140 characters. In which case, the tremendous data it produced can help addressing issues oriented from various applications (Yan, Yang, & Wang, 2017). One of them is sentiment analysis. Researchers have used the Twitter data to predict political sentiment and people's political preference (Ceron, Curini, Iacus, & Porro, 2013) (Tumasjan, Sprenger, Sandner, & Welpe, 2010). Sentiment analysis is to predict user's emotion through the tweet information he wrote.

Twitter sentiment analysis is usually considered as a supervised classification issue. In this project, tweets are commonly categorized into ten classes: clap, cry, disappoint, explode, facepalm, hands, neutral, shrug, think, upside. Each of them corresponds to an emoji expression. This report briefly explains the two data mining methodologies used with Weka (Eibe, Mark, & Ian, 2016), and then generally discusses about the model built to predict emoji classes for testing data.

2. Data Set

The data used is obtained through Twitter API[2] by the teaching team. This report used the *train_most100.arff* to build the model and verified with the *dev_most100.arff* file. During the process, this report produced *train_NB.arff, dev_NB.arff, train_J48.arff,* and *dev_J48.arff* files with refined attributes. These are the files used to train the model.

3. Data Mining

In this report, two data mining methodologies are used. One is Naïve Bayes and the other is Decision Tree.

3.1 Utilized Algorithms

3.1.1 Naïve Bayes

The Naïve Bayes is an algorithm based on the Bayes' theorem and the total probability (Patil & Mrs. S. S. Sherekar, 2013). In this report, the author used the Naïve Bayes classifier to build the first model. This classifier is a simple probabilistic classifier calculating a set of probabilities through counting the frequency and combinations of values in the given data set. It can have best performance when features used is independent.

3.1.2 Decision Tree

Another methodology used in this report is the J48 classifier. It is a simple C4.5 Decision Tree for classification. While creating a binary tree, the classifier ignores the missing values and predict them based on what is known about the attribute values for other records. The basic idea is to divide the data into range based on the attribute values in the training dataset.

3.2 Original Attributes Analysis

In the original data set, 102 attributes are provided including id and emoji classes. In this report, the author firstly implemented two classifiers with original attributes and default parameters. The file *train_most100.arff* is used to train the model and *dev_most100.arff* is used for validation. Some evaluation metrics are used to compare the performance of two classifiers, including accuracy, precision and recall. The result is shown in the following table.

[1] Twitter, Inc. https://twitter.com/

[2] https://developer.twitter.com/en/docs/tweets/search/api-reference/ get-search-tweets

***Table 1**: Results for NB & J48 with Original Attributes*

	Naïve Bayes(NB)		J48 DecisionTree	
Accuracy	30.3765%		45.3469%	
runtime	0.57s		61.91s	
Class	Precision	Recall	Precision	Recall
Clap	0.245	0.534	0.540	0.530
Cry	0.361	0.294	0.497	0.459
Disappoint	0.191	0.141	0.291	0.221
Explode	0.459	0.261	0.439	0.549
Face Palm	0.232	0.148	0.395	0.286
Hands	0.629	0.378	0.803	0.673
Neutral	0.250	0.235	0.316	0.429
Shrug	0.208	0.184	0.389	0.333
Think	0.298	0.385	0.485	0.451
Upside	0.279	0.279	0.359	0.356
Avg.	0.330	0.304	0.464	0.453

As above table shows, it is clearly that the J48 classifier has a higher accuracy than Naïve Bayes classifier for this data set. One probable reason for this is that the attributes may not be absolutely independent to each other, which will lead the NB classifier to a bias and decrease the accuracy.

The runtime for two classifiers building the model are also presented in the above table. J48 classifier takes extremely longer time to build its model than NB classifier. The reason for this maybe because the number of attributes is large, which will take a long time to build the tree. Also, the calculation of NB classifier is much simpler.

The precision and recall of each class in J48 results are commonly greater than those in NB classifier's results. One exception is the precision value for *Explode* class. This means that the NB model's prediction for Explode class is more trustable than J48 model. That is, for *Explode* class, the Naïve Bayes classifier has a better performance than J48.

Similarly, as the red values shown above, the precision and recall values are higher than other emoji classes, which means that both NB and J48 classifiers have a better performance in predicting the *Hand* class.

3.3 Refine Attributes

As mentioned above, one probable reason for the low accuracy is that some attributes are not independent and can hardly express sentiment. For instance, the attributes like "id", "an", "and" etc. have no attributes to emoji classification. Therefore, this report picked following possible unrelated attributes, then delete each of them and observe if it has impact on the accuracy of two classifiers. The results are presented in the below table2. Red values mean the accuracy is increased according to the original one.

***Table 2**: Accuracy after deleting each possible unrelated attribute*

%	**Original**	id	about	an
NB	30.3765	30.3354	30.5656	30.3930
J48	45.3469	46.1361	45.4785	45.2647
	and	are	as	at
NB	30.6067	30.6396	30.3930	30.7793
J48	45.3291	45.3391	45.2647	45.5278
	be	follow	for	in
NB	30.3847	30.7547	30.3108	30.5081
J48	45.1989	45.3223	45.1496	45.2551
	is	it	me	of
NB	31.4946	30.5245	29.9244	30.8533
J48	44.9688	45.3018	45.1578	45.3469
	on	one	or	our
NB	30.2121	30.3930	30.2039	30.7300
J48	45.3305	45.3100	45.1332	45.7004
	out	people	they	time
NB	30.3354	30.3930	30.3272	29.9819
J48	44.9523	45.3963	45.1578	44.9277
	today	was	we	when
NB	30.5081	30.4998	30.4012	30.3025
J48	45.4045	45.0592	45.1332	45.0592
	who	will	with	you
NB	30.2039	30.4505	30.6807	30.4505
J48	45.6347	45.3058	45.1661	44.5824
	your			
NB	30.3847			
J48	44.9688			

After deleting the unrelated attributes for two classifiers, the refined training data sets are saved as *train_NB.arff* and *train_J48.arff*. The number of refined attributes for NB model is 81, and for J48 model is 97. The accuracy of two models trained with the refined attributes are presented in the following table 3 compared with the original one.

***Table 3**: Accuracy with refined attributes*

	Original	Refined
Naïve Bayes	30.3765%	31.6590%
J48 Decision Tree	45.3469%	46.3745%

3.4 Pruned Decision Tree

Besides refining attributes can optimize the models, pruning tree also will gain a higher accuracy. It is an important step to reduce tree size and overfitting, which will optimize the performance for a decision tree. Two parameters control pruning tree:
-C, confidenceFactor (default is 0.25) and
-M, minNumObj (default is 2).

Using the refined data, the results of accuracy comparing unpruned and pruned tree are shown in table 4.

***Table 4**: Accuracy of unpruned and pruned tree (-C = 0.25)*

minNumObj	Unpruned	Pruned
2	45.5853%	46.3745%
5	44.2946%	44.8372%
10	43.2012%	43.4808%

As shows above, model will gain a higher accuracy after pruning the decision tree. In detail, accuracy changes according to the value of two parameters.

***Table 5**: Accuracy changing -C and -M to prune a tree (%)*

Confidence Factor	minNumObj		
	2	5	10
0.01	46.3828	44.7879	43.1519
0.1	46.5143	45.125	43.9247
0.25	46.3745	44.8372	43.4808
0.3	46.2923	44.7139	43.5301

As presented in the above table, with the increment of confidence factor, the accuracy fist increases then decreases. It is because smaller confidence factor indicates greater estimated error rate to a node, which needs more pruning and finally leads to over-training and lower accuracy.

minNumObj means minimum number of instance per leaf. With the increment of minNumObj, accuracy is getting smaller. This is because: when the threshold of not splitting nodes is greater, the model will have less complexity and won't overfitting.

4. Conclusion

After optimizing the performance, the final model used for testing has an accuracy of 46.5143%. It is a J48 classifier trained with the refined dataset and parameters of -C 0.1 -M 2. This model is still not perfect. One reason is that the most 100 words may not be the most words describing emotions. Hence, I believe changing features may gain a better performance.

5. Bibliography

Ceron, A., Curini, L., Iacus, S. M., & Porro, G. (2013). Every tweet counts? How sentiment analysis of social media can improve our knowledge of citizens' political preferences with an application to Italy and France. *New Media & Society* , 340 - 358.

Eibe, F., Mark, A. H., & Ian, H. W. (2016). *The WEKA Workbench. Online Appendix for "Data Mining: Practical Machine Learning Tools and Techniques".* Morgan Kaufmann.

Patil, T. R., & Mrs. S. S. Sherekar. (2013). Performance Analysis of Naive Bayes and J48 Classification Algorithm for Data Classification. *International Journal Of Computer Science And Applications*, 256-261.

Tumasjan, A., Sprenger, T. O., Sandner, P. G., & Welpe, I. M. (2010). Predicting Elections with Twitter: What 140 Characters Reveal about Political Sentiment. *Icwsm*, 178-185.

Yan, Y., Yang, H., & Wang, H.-M. (2017). Two simple and effective ensemble classifiers for Twitter sentiment analysis. *2017 Computing Conference*, 1386-1393.