Measuring Emotions from Online News and Evaluating Public Models from Netizens' Comments: A Text Mining Approach

Simon Fong
Department of Computer and Information Science
University of Macau
Taipa, Macau SAR
Email: ccfong@umac.mo

Abstract— **Nowadays netizens embark on a prevalent lifestyle to actively voice out their opinions online that includes both forums and social networks (Web 2.0). Their opinions which initially are intended for their groups of friends propagate to attentions of many. This pond of opinions in the forms of forum posts, messages written on micro-blogs, Twitter and Facebook, constitute to online opinions that represent a community of online users. The messages though might seem to be trivial when each of them is viewed singularly; the converged sum of them serves as a potentially useful source of feedbacks to the current affairs after analysis. A local government, for instance, may be interested to know the response of the citizens after a new policy is announced, from their voices collected from the Internet. However, such online messages are unstructured in nature, their contexts vary greatly, and that poses a tremendous difficulty in correctly interpreting them. In this paper we propose an innovative analytical model that evaluates such messages by representing them in different moods. The model comprises of several data analytics such as emotion classification by text mining and hierarchical visualization that reflects public moods over a large repository of online comments.**

Index Terms— **Emotion classification; text mining; hierarchical visualization**

I. Introduction

Netizens nowadays develop a habit of whining out their opinions in the virtual world, through blogs, social networks as well as community forums. Their purpose may be just to share their views, casually or consciously, in response to all kinds of world events and individual topics of interest. From the postings and counter-replies, it has evolved into a trend of social acquaintance in the virtual world [1]. Twitter has more than 180 million unique visitors per month, and a total amount of messages close to a trillion. Facebook also has a population of 166 million active users whose posts amount to an astronomical figure. And these figures are still undergoing some phenomenal growth.

Recently government agencies established their community groups on Facebook. The motive could be in twofold: to disseminate information to the online users, and to probably listen to their opinions. However, to the second motive, assuming that the government agency bothers to pay attention to those opinions, there is an inherent challenge in the format of the data. They are unstructured both in grammar and context. As users are free to post anything under the sun, the format is not in formal writing (unlike official letters); slangs may be used and they differ from culture to culture. On the brighter side, netizens are responsive to new posts and new events. For example, any world news, such as earthquake, terrorist attacks or economic crisis that rock the world would attract them to proactively post and encounter post on each other's messages. They share their views in different emotions, pertaining to the subject that they are commenting about. The online messages come in very different types of wish-making, suggestion, political opinion, critics and praises, or dissatisfaction to share among friends and the rest of the world.

In addition to the obstacles of data formats and contexts, a government or organization may face another challenge due to the dynamic nature of the distributed online comments, which arise both in tremendous quantity and at a very high speed. The contents of the comments may change over time too; for example, an invention of a vaccine for a global epidemic disease may first be cheered as "happy" news. Should it be later found as a hoax, the general comments may gradually switch to mood of "disappointing" or even "hilarious".

Organizations do need some autonomous method to classify the messages into different moods and kinds of opinions, in contrast of the previous works of deciphering their actual meanings. Currently manual work is required by a human user to comprehend the messages by his knowledge background and relate them as opinions being talked about of a particular event. Because there are large diversities of words and vocabularies representing different emotions, it is important to tap on the cultural background knowledge.

Emotion is a complex psycho physiological experience of an individual's state of mind as interacting with biochemical and environmental influences. In humans, emotion fundamentally involves "physiological arousal, expressive behaviors, and conscious experience" [2]. Emotion is associated with mood, temperament, personality and disposition, and motivation. Emotion doesn't exist in computers that are based on logics. Emotion also may not be easily calculated by a formula.

doi:10.4304/jetwi.4.1.60-66

Emotion is a fuzzy state; hence machine learning algorithms that are able to represent non-linear relations between the occurrence of a series of keywords in the text and the predicted class of emotion, are appropriate for handling this type of problems such as Artificial Neural Network and Decision Tree algorithms [3]. Before entering the comments in to an emotion classifier, we need to translate sentences to relative metadata which are represented in abstract levels. To translate sentences to metadata we make use of a linguistic dictionary to categorize, stemming methods that filter out unimportant words, vector space model for establishing the importance of words by measuring their frequencies and group the significant words into meta-data.

"Point of view" is another important factor that contributes to understanding emotions. We utilize information from online news to establish a neutral evaluation standard. Since opinions in newspapers are in journalistic and relatively objective style, we adopt so as a standard for describing neutral opinions. The other usage of newspaper is it may contain the background story of an event. By comparing the evaluation standard and online comments we can have some benchmark for positioning a neutral point in our visualization which shows information in different levels, thus we call it "Hierarchical Visualization". The Hierarchical Visualization can provide the trend of public mood, detail of the range about mood and it can be directly used for government or organization to understand their citizens' or customer's feedbacks. The Hierarchical Visualization reveals the moods of the public in general, instead of displaying a long list of individual comments. Our proposed Hierarchical Visualization is designed for high-level users who often prefer to glimpse at an overall view of the public opinions, without going into details or crunching over the numeric figures. This method proposed in this paper is subtle which doesn't require costly massive scale of survey questionnaires that probe answers directly from citizens.

II. Our Proposed Model

Before collecting information from the Internet, a specific Research Topic (or topic of interest) to be analyzed should be defined. Research topics could be of any current affair or any latest government policy which netizens are keen to comment about. The next step is to download the data from relevant sources. The easiest way to confirm the date of the event that happened can be referred from the official news. News published on the Internet usually would have highlighted by some keywords that can be extracted from the tags that appear at the bottom of the page – the keywords are useful for us to define the metadata of a research topic. Overall, for each research topic, we use the time, the metadata, as settings of parameters for the web downloading software to congregate Internet comments within a reasonable time range (e.g. 80% of netizens talked about Michael Jackson's death within only 5 months). The information downloaded will be used to build up two kinds of databases. One is a repository of online News that are tagged with date of occurrence, plus the related metadata for ontology [4]; the second one is the postings extracted from some social networks and micro-blogging sites. Twitter and Facebook are used as experiments in this paper. HTML tags are cleansed in the preprocessing step. The information about the poster's information, such as IP (which may not always be available), time of posting, user's background or other will also be collated.

There are several approaches to build a Moods engine which is used to classify the mood of a given online article or piece of text. It was suggested in [5] that a Moods engine to be based on a standard dictionary for embracing the keywords by using an artificial neural network (ANN). The relevant words that are related to different emotions from a well-known dictionary reference are used as training data to build up a number of ANNs, one for each type of emotion so that it can be used subsequently to recognize the perceived emotion out from a testing text. Optionally, one may incorporate MSN-style of acronyms or emoticons to represent emotions [6], e.g. a smiley is a symbol of happiness written as :-) Short-names commonly used as cyber etiquettes like W.T.H/F. (anger plus astonishment) and I.M.H.O (neutral narration) could also be added on. One ANN is to be trained and employed to describe one type of mood. The mood engine is to be fine-tuned with users' subjective experiences for improving the accuracy. So the major function of the mood engine is to distinguish a resultant mood by reading through a pile of text messages.

The evaluated news will also train the ANN of different moods. The ANN of a particular mood is represented by the weights trained by the training data. If a piece of news was marked or flagged as "happiness", the news would be used to train the "happiness" ANN model. Data from the Internet comments databases would be used as testing data to be tested in the ANNs for deciding which mood(s) they belong to.

Alternatively, as proposed in this paper here, a generic text classifier can be trained by some predefined training texts which have already the labeled classes of emotions assigned in the training data. This approach requires pre-assignment of verdict classes (emotions) on the training dataset that is made up of news which we already known the emotion class that they belong to. The training dataset would be processed by text mining techniques that include a sequence of data-preprocessing steps such as stemming, data cleaning and transforming of the keywords to attributes of frequency of occurrences. Text classification is adopted here as a generic approach that could be powered by a range of different underlying algorithms. After a classifier is trained with sufficient records that have the labeled emotions, it could be deployed for automatically classifying the expected emotions from the future texts. It is recommended that the classifier has to be trained first to certain acceptable accuracy from training data taken from online news pertaining to a Research topic or a category of current affairs. Then it would be used for classifying emotions from testing data that are to be scraped from users' posts. The workflow of the model training is shown in Figure 1.

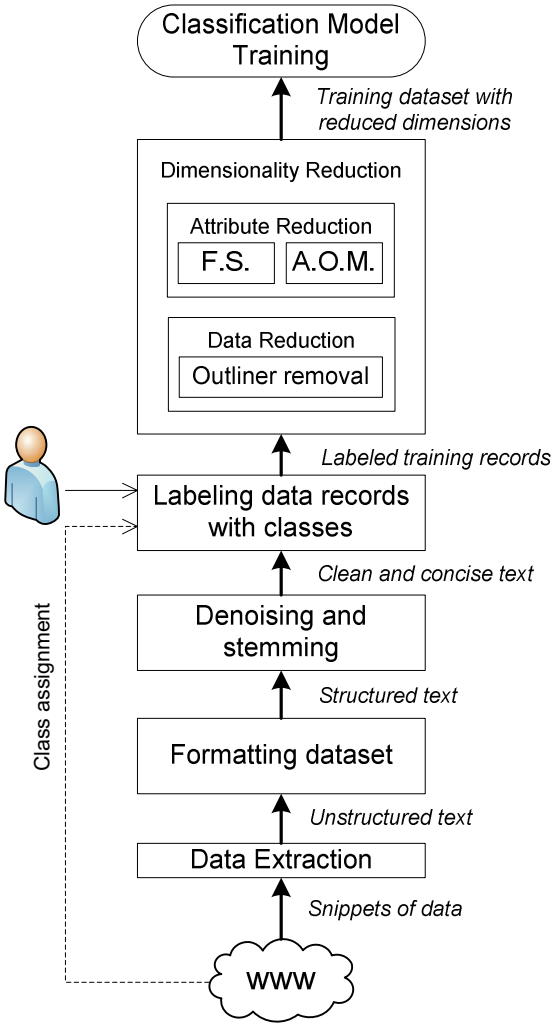

Figure 1. The workflow of training the classifier for recognizing emotions from online news.

Once the emotion classifier is trained and its performance accuracy reaches an acceptable level, the classifier is ready to classify emotions from new data. Posts and messages made by online users are retrieved and formatted in the same way as in the training process for the classifier. Given the new data, the classifier classifies new instances of online users' opinions into emotion groups. The advantage of this text mining approach is the generality that different machine learning algorithms as well as different dimensionality reduction algorithms can be used, even in combination, for the optimal results. Figure 2 shows the subsequent process.

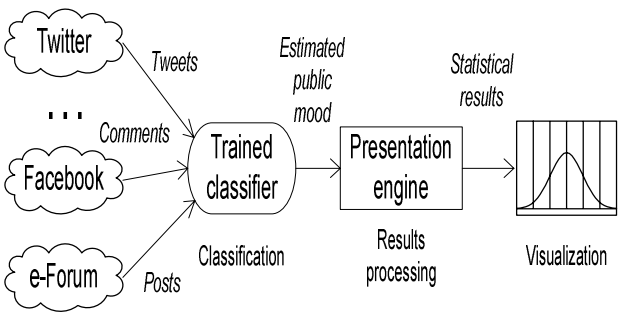

Figure 2. High-level view of the emotion classification process.

The output results will be processed in presentation engine and displayed in the Hierarchical Visualizations. Multiple levels of visualizations are used because the details of the results could be shown in different depths, depending on the choice of the user for the desired resolution. Too much information in visualization is confusing to the users. Users can opt to choose a viewing level interactively in the control panel of the visualization software.

The results are shown in graphical form so that it is easy to captivate the attention of the user, and possibly to spot any special patterns visually. The user can zoom in and out at will, or to display the full details for further analysis when necessary. Colors are used to represent the different emotions respectively. The following example in our experiment is on the topic of "Hengqin Campus Project of University of Macau". We set up a list of colors [7] to represent different moods. The circle represents moods of an event and we use the angles represent the percentage of the moods. In Figure 3, it is easy to get to know the public mood about an event based on the comments collected from the Internet. In the software, we can scale to level 2 of the hierarchical visualization as shown in Figure 4 when we select the two colors in the circle. Since the colors are not fixed, the visualization is interactive with the end-user. The user can focus his attention on which moods that the general public feels about the specific topic. In this zoom-in level of visualization, we may want to analyze about the genders of the users who posted their opinions (and subsequently reflected by their moods), just for example. The wave line in Figure 4 is representing the number of people who are in different moods with different gender, one on each side of the belt. In the next level, we can select a location to be analyzed. The locations, gender and moods relationships are presented in this level. In Figure 5, the chart shows visually that how users in different locations carry certain moods. The level of details can be increased optionally; for example, the locations can further break down to suburbs, streets, etc. Other dimensions can be added or switched too

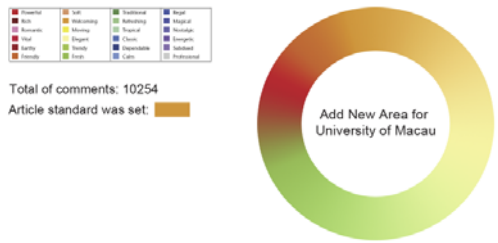

Figure 3. Level 1 of the visualization.

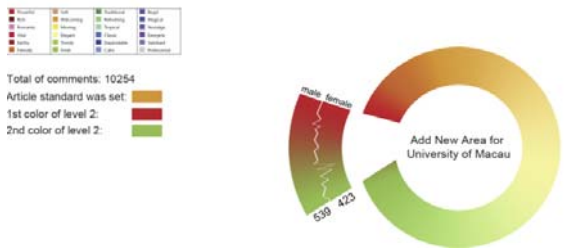

Figure 4. Level 2 of the visualization.

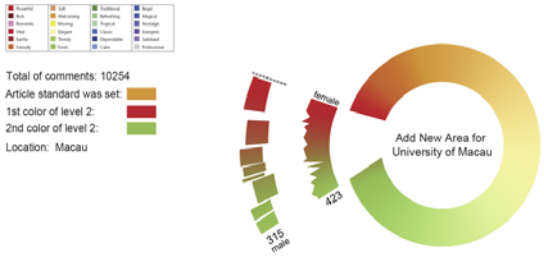

Figure 5. Level 3 of the visualization.

The proposed system can help organizations or government to understand the opinions which are in response to a news event or a policy announcement, without doing meticulous survey to collect public opinions. The system is easy enough to use, for revealing the public moods based on a given event. The system features about consideration of a culture of a country by training the emotion classifier using news samples from the local online news website. For example, CNN for Americans, CBS/BBC for English, ABC for Australian, ChannelNewsAsia for Asians, just to name a few. Cultural perceptions influence on how the citizens express their views, hence the choice of words being said and posted on online forums. For example, the word "cool" may mean a cheerful mood in the Western culture, but otherwise in oriental or other conservative cultures who take the word "cool" literally as "indifferent" in character. However, different versions of the system may be needed to be built for different cultures [8], but the training datasets and hence the classifiers would be unique in each culture. As such, an analyst who uses the system by different cultures can understand where, who, which age group, how many people and what they feel in the visualization, in response to an event, based on the analysis from the Internet comments.

III. EXPERIMENT

In order to validate the concept of our proposed model, a text mining program is built by using Weka which is an open-source JAVA platform for evaluating machine learning algorithms by University of Waikato. Specifically, text classification is implemented under the text mining domain and training datasets of different emotions are used for the experiment. We aim to test-drive the classifier with different machine learning algorithms and different dimensionality reduction methods. It is a known challenge in text mining that the accuracy is almost directly pegged on how well the dimensionality of the dataset can be tamed. The training data which are obtained from online news platforms are unstructured in nature. In addition to standard data pre-processing techniques like filtering noise and stemming (a process for removing redundant words), dimensionality reduction algorithms for reducing the number of attributes that are used to represent the essence of the text and amount of instance number are applied in our experiment. An outliner removal algorithm is used for trimming off data rows that have exceptionally different values from the norm. For reducing the number of attributes, two standard Feature Selection algorithms (FS) are used, together with a novel approach called Attribute Overlap Minimization (AOM) are applied. Readers who want to have further details about these algorithms should refer to [9].

The training data are excerpted from CNN news website, of the news articles that were released for ten days across the New Year 2012 (one week before and one week after the New Year eve). The news collection has a good mix of political happenings, important world events and lifestyles. One hundred sample news were obtained in total, and they were rated manually according to six basic human psychological emotions, namely, Anger, Fear, Joy, Love, Sadness and Surprise. The data are formatted into ARFF format (as required by Weka), having one news per row in the following structure: <emotion>, <"text of the news"> where the second field has a variable length. The training dataset is then subject to the above-mentioned dimensionality reduction methods for transformation to a concise dataset in which the attributes have substantial predictive powers contributing to recognizing emotions from the text strings.

The Feature Selection algorithms used include, *CfsSubset*, *ChiSquaredAttribute*, *InfoGainAttribute*, *SignificanceAttribute*, and *SymmetricalUncertAttribute*. The full explanation about these algorithms can be found on Weka homepage. As shown in our experimental results in Table 1, *CfsSubset* generally can achieve the best classification accuracy by filtering most but retaining only the minimum set of elite attributes that have most predicting powers. The rest of the FS algorithms produce almost identical results though *ChiSquare* and *InfoGain* are relatively more popularly used in data mining community. Accuracy is defined by the percentage of the number of correctly classified instances over the total number of instances in the training dataset. By applying attribute reduction and data reduction, we can observe that the initial number of attributes have reduced greatly from 8135 to 19. Having a concise and elite amount of attributes is crucial in real-time application, and in text mining, the number of attributes is proportional to the coverage of news articles – the more unique words (vocabularies) that are being covered, the greater the amount of attributes there are. Text classification essentially works on the principle of finding the non-linear relations of co-occurrences of important keywords in a given text, measured by their occurrence frequencies.

It is found from the results in Table 1 that using the FS algorithm *CfsSubset* together with other techniques can effectively achieve a classifier (Decision tree is selected in this example) that has the lowest number of tree size, highest accuracy and shortest training time. A compact tree size in Decision tree algorithm means least consumption of heap memory space that is essential for real-time applications where memory space may be an operational constraint. Short training time implies that the classifier model takes only a short while for updating or even rebuilding the tree model that will be useful for application scenarios where frequent updates may be necessary for fast-changing data inputs.

The experiment is then extended to evaluate the use of machine learning algorithms, with the objective of achieving the highest accuracy. The selection list of the machine learning algorithm used in our experiment here is by no means exhaustive, but will form the basis of a performance comparison which should supposedly cover most of the popular algorithms. The machine learning algorithms are grouped by four main categories, Decision Tree, Rules, Bayes, Meta and Miscellaneous; all of them are known to be effective for data classification in data mining to certain extents. The list of algorithms is shown in Table 1, and their details can be found in [9].

TABLE I. LIST OF CLASSIFICATION ALGORITHMS USED IN THE EXPERIMENT

Acronym	Algorithm name	Type
J48	C4.5 decision tree	Decision tree
BFTree	Best-first decision tree classifier	
Ftree	Functional trees', which are classification trees that could have logistic regression functions at the inner nodes and/or leaves	
NBTree	A decision tree with naive Bayes classifiers at the leaves	
LMT	Logistic model trees', which are classification trees with logistic regression functions at the leaves	
RandForest	A forest of random trees	
RandTree	A tree that considers *K* randomly chosen attributes at each node	
REPTree	Fast decision tree learner. Builds a decision/regression tree using information gain/variance and prunes it using reduced-error pruning	
DecTable	A simple decision table majority classifier	Rules
FURI	Fuzzy Unordered Rule Induction Algorithm	
Ripper	A propositional rule learner, Repeated Incremental Pruning to Produce Error Reduction (RIPPER)	
PART	A PART decision list. Uses separate-and-conquer. Builds a partial C4.5 decision tree in each iteration and makes the "best" leaf into a rule	
BayesNet	Bayes Network learning using various search algorithms and quality measures	Bayes
CompNB	A Complement class Naive Bayes classifier	
NB	a Naive Bayes classifier using estimator classes	
Bagging	Bagging a classifier to reduce variance	Meta
Ensemble	Combines several classifiers using the ensemble selection method	
SVM	Support Vector Machine	Misc
NN	Backpropagation Neural Network	

The experiments are conducted according to the workflow depicted in Figure 1. After the training data are cleansed, formatted and labeled, they were subject to the dimensionality reduction algorithms for improving the accuracy of the classifier. Three groups of resultant datasets were text-mined by different classification algorithms that are specified in Table 1. They are the original dataset without any dimensionality reduction, transformed dataset with reduced attributes, and transformed dataset with both attributes reduced and outliners removed. The results in terms of accuracy are shown in Figure 6 and Figure 7. The experiments are repeated for trying two most popular FS algorithms – *CfsSubset* and *InfoGain*. While the former FS algorithm chooses only the minimum number of the attributes that have significant contributing predictive power, the latter algorithm retains most of the attributes that have at least non-zero information gain towards the decision tree induction. Figure 6 shows the classification results from dataset filtered by *CfsSubset*, and Figure 7 shows those filtered by *InfoGain*.

It can be seen that the classifiers performed poorly over the original dataset, but the accuracy greatly improved once the attributes are reduced. A more than 100% gain increase is observed between the original data and the attribute-reduced data. This gain can be observed for most of the classification algorithms except for FURI and RIPPER. It makes little difference between FS algorithms for *CfsSubset* and *InfoGain*, which means *CfsSubset* can be used for minimum number of attributes. From the results of Figures 6 and 7, approximately 4% to 18.3% increases in accuracy are observed between the results obtained from attribute-reduced data and both attribute-reduced and outliner removed data. All the classification algorithms perform consistently well. Naïve Bayes classifier (NB) however, achieves the highest accuracy 86.8% in all cases. It in fact is the only classifier which has no effect by using *CfsSubset* and *InfoGain* feature selection algorithms. NB is independent of the feature selection algorithm and the candidate that yields the highest accuracy when a combination of dimensionality reduction techniques is used. For this reason, mood classification application is suggested to adopt NB for effectively classifying online text messages into a class of one of the six emotions. The following part of the experiment is to classify newly acquired text data by using the trained classification model.

IV. RELATED WORK

There is a similar project named "Twitter mood maps reveal emotional states of America" in America. It has an idea to present human mood in a timeline superimposed on an America map with different colors. This method takes individual words out of context. If someone tweets "I am not happy", the team's method counts the tweet as positive because of the word "happy". It is based on current state of Twitter user and by possibly keyword matching methods. Unlike our proposed model, it does not shows results from multi-dimensions, and the fuzzy

natures of mood matching and different cultural aspects were not considered.

Another academic research work that is very similar to ours is [10], by the authors Tao et al. Tao regarded that the Web has become an excellent source for gathering and realizing public voice. The paper discusses a method for exploring the public mood levels at the time of posting. Hence the results are presented as two-dimensional graphs with the y-axis being the mood level, and the x-axis as a time-line. Again, the two dimensions of variables for presenting mood levels can be extended to multiple dimensions as proposed in this year. Also the paper [10] used corpus aggregating method for measuring mood level, and the case study was on emergency scenarios, where a flexible classifier that can be chosen from a collection can be used in our model for handling application situations. In addition, our model incorporates with a hierarchical visualization program, and our experiments showed that the prototype can be used in general situations.

V. CONCLUSION

In this paper we proposed and defined an analytical model for evaluating online users' comments in response to a given event. Our model features a Mood engine made up of a number of dimensionality reduction algorithms, and text classification machine learning algorithms, that can effectively classify a text into one of the six human basic emotions aka moods. The moods can be changed and calibrated according to different cultures by re-training the text classification model. The classifiers can be trained by using standard words or past news with predefined emotions assigned by human experts. The trained classifier is then used to detect types of moods and their intensities from a pool of new messages and postings collected from micro-blogs and social networks that constitute a large online community as a whole. The online comments are inputted to the mood engine and the comments are categorized into types of moods. Aggregating categorized comments and their moods are fed into a hierarchical visualization program that shows interactively different dimensions of the information with respect to the public Mood. A prototype is built and results show that the model is feasible. This paper contributes a text classification model for this job; its efficacy is experimented by considering a wide range of classification algorithms and several feature selection algorithms. It is found that Naïve Bayes classification algorithm outperformed the rest, and it is independent of which feature selection algorithm is being used. The techniques and model presented in this paper are generic which means a specific algorithm can be replaced by a new candidate, and it is believed to work on other similar mood detection scenarios such as analysis of help-desk logs, customers' feedbacks, online reviews etc. The core engine of the model is the Mood Engine which essentially is shown to be possible by implementing it with an appropriate text mining algorithm and a sequence of dimensionality reduction methods. The advantage is a simple and flexible model with reasonable accuracy.

REFERENCES

[1] Tanase, S., "When web 2.0 sneezes ...everyone gets sick", Engineering & Technology, Volume 5, Issue 5, 27 March-23 April 2010, pp.28-29.

[2] Myers, David G. "Theories of Emotion." Psychology: Seventh Edition, New York, NY: Worth Publishers, 2004, p.500 (Wiki).

[3] Elaleem, O.A.; Elragal, H.M.; Shehata, H.M, "Voice Message Priorities Using Fuzzy Mood Identifier", Proceedings of the Twenty Third National Radio Science Conference, NRSC 2006, pp.1-6.

[4] Seheon Song; Minkoo Kim; Seungmin Rho; Eenjun Hwang, "Music Ontology for Mood and Situation Reasoning to Support Music Retrieval and Recommendation", Third International Conference on Digital Society, ICDS 2009, pp.304-309.

[5] Chan Io Weng, Simon Fong, Suash Deb, "An Analytical model for Evaluating Public Moods Based on Internet Comments", National Conference on Data Mining (NCDM 2011), 19-20 February 2011, Pune, India.

[6] Yamashita, Ryo; Yamaguchi, Sanae; Takami, Kazumasa, "A Method of Inferring the Preferences and Mood of Mobile Phone Users by Analyzing Pictograms and Emoticons Used in their Emails", Third International Conference on Advances in Human-Oriented and Personalized Mechanisms, Technologies and Services (CENTRIC), 2010, pp.67-72

[7] Chuan-Kai Yang, Li-Kai Peng, "Automatic Mood-Transferring between Color Images", IEEE ComputerGraphics, Issue No.2, March/April 2008, pp.52-61

[8] Zhijun Zhao; Lingyun Xie; Jing Liu; Wen Wu, "The analysis of mood taxonomy comparision between chinese and western music", 2nd International Conference on Signal Processing Systems (ICSPS), 2010, pp.606-610

[9] Simon Fong, "Dimensionality Reduction in Vector Space Model for Multi-class Text Classification", submitted to JIPS-K, 2012.

[10] Tao Xu, Qinke Peng, Chengwei Li, "A Method of Capturing the Public Mood Levels in Emergency Based on Internet Comments", 7th World Congress on Intelligent Control and Automation, WCICA 2008, 25-27 June 2008, pp.3496-3499.

Simon Fong

He graduated from La Trobe University in Australia, with a First Class Honours BEng. Computer Systems degree and a PhD. Computer Science degree in 1993 and 1998 respectively. Simon is now working as an Assistant Professor in the Computer and Information Science Department of the University of Macau. He is also one of the founding members of the Data Analytics and Collaborative Computing Research Group in the Faculty of Science and Technology. Before his academic career, Simon took up various managerial and engineering posts, such as being a systems engineer, IT consultant, integrated network specialist, and e-commerce director in Melbourne, Hong Kong. and Singapore. Some companies that he worked at before include Hong Kong Telecom, Singapore Network Services, AES Pro-Data, and the United Overseas Bank in Singapore. Dr. Fong has published over 150 peer-reviewed international conference and journal papers, mostly in the area of E-Commerce and Datamining.

Accuracy

Classifiers	Original	Attr-Reduced	Attr-Data-Reduced
J48	33	66	75.8242
BFTree	32	64	72.5275
Ftree	36	78	83.5165
NBTree	31	69	76.9231
LMT	37	78	83.5165
RandForest	33	75	83.5165
RandTree	33	75	83.5165
REPTree	34	62	71.4286
DecTable	26	60	71.4286
FURI	38	55	65.9341
Ripper	32	55	63.7363
PART	31	65	76.9231
BayesNet	22	72	79.1209
CompNB	36	68	71.4286
NB	46	83	86.8132
Bagging	37	71	74.7253
Ensemble	34	64	69.2308
SVM	38	69	67.033
NN	36	75	82.4176

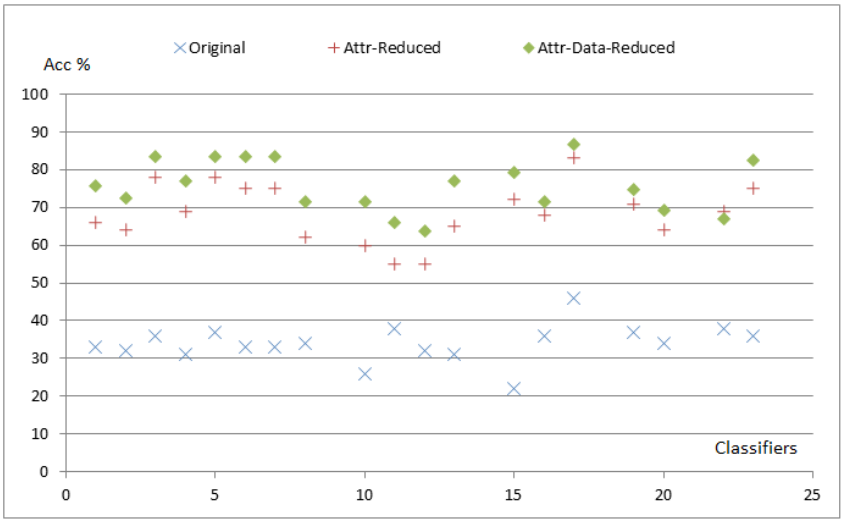

Figure 6. Accuracy results of classifiers in percentage with different types of datasets based on *CfsSubset* FS algorithm.

Accuracy

Classifiers	Original	Attr-Reduced	Attr-Data-Reduced
J48	33	69	75.8242
BFTree	32	66	73.6264
Ftree	36	73	79.1209
NBTree	31	69	73.6264
LMT	37	73	78.022
RandForest	33	76	80.2198
RandTree	33	76	78.022
REPTree	34	60	71.4286
DecTable	26	58	70.3297
FURI	38	60	62.6374
Ripper	32	56	62.6374
PART	31	67	75.8242
BayesNet	22	71	74.7253
CompNB	36	72	73.6264
NB	46	83	86.8132
Bagging	37	68	74.7253
Ensemble	34	62	71.4286
SVM	38	63	63.7363
NN	36	75	81.3187

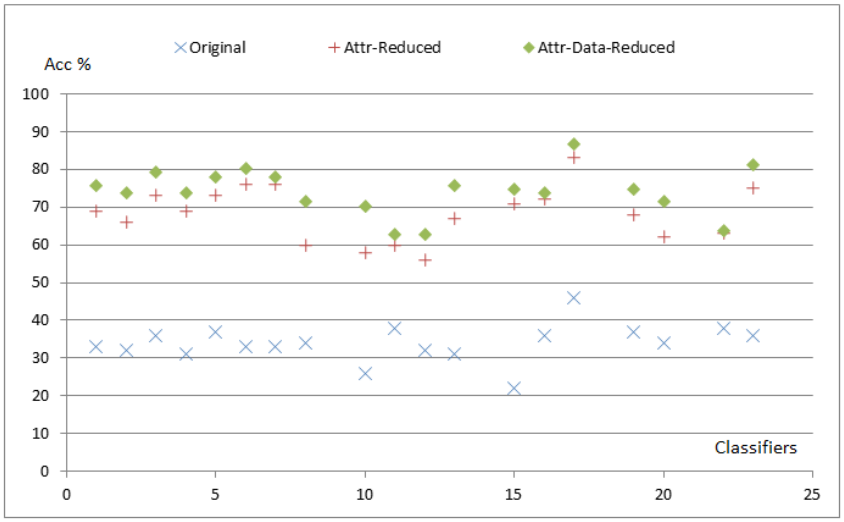

Figure 7. Accuracy results of classifiers in percentage with different types of datasets based on *InfoGain* FS algorithm.