KNOWLEDGE ENGINEERING: PRINCIPLE AND TECHNIQUE, KEPT 2009
International Conference on Knowledge Engineering Principles and Techniques
Selected Papers, Cluj-Napoca (Romania), July 2-4
2000

ANALYSIS AND GENERATION OF EMOTION IN TEXTS

DIANA INKPEN, FAZEL KESHTKAR, AND DIMAN GHAZI

Abstract. This paper explores the task of automatic emotion analysis and generation in texts. We present preliminary results for the task of classifying texts by classes of emotions. Then, we present detailed experiments in classifying texts by classes of mood. We propose a novel approach that uses the hierarchy of possible moods in order to achieve better results than a standard flat classification. We also show that using sentiment orientation features improves the performance of classification. At the end, the possibility of generating texts that express specific emotions is discussed.

1. Introduction

Automatic text classification is usually done by using the topics of the documents as classes. In this case, the words in the documents are very good indicators for each class. It is more difficult to classify text by genre, style, author, or sentiment orientation, because the classification has to be done regardless of the topic of the document.

The automatic detection of emotions in texts is important for applications such as: opinion mining and market analysis, affective computing, natural language interfaces, and e-learning environments, including educational games.

We also want to use the results of emotion analysis in order to automatically generate text that expresses emotions.

Since we discuss classification by emotions and by mood, we need to clarify the difference between the two terms. Emotions are momentary changes that influence the text written by a person, while moods are medium-term states of a person, which can be shifted by the emotions that are expressed.

This paper is organized as follows: Section 2 briefly presents related work. Section 3 presents preliminary experiments in emotion classification. Section 4 explains our hierarchical method for mood classification. Section 5 discusses the possibility of generating texts with emotions.

2. Related work

On top of the large body of work on automatic text classification by topic, a lot of progress has been done in opinion and sentiment analysis [13].

Research on emotion and mood detection is just starting. Holzman [6] and Rubin et al. [15] investigated emotion detection, but on very small data sets. There is recent work on classifying sentences from blogs [2] and newspaper headlines [17] into six classes of emotions [5]. Classification was also done into nine classes of emotions [7], for sentences from blogs [12] and on sentences from fairy tales [1].

Very few researchers studied mood classification. Mishne [11] collected a corpus of blog data annotated with mood labels, and implemented a Support Vector Machine (SVM) classifier. He used features such as frequency counts, lengths, sentiment orientations, emphasized words, and special symbols, and classified blogs into the 40-most frequent moods. Another mood classification system was proposed by Jung [8], using some common-sense knowledge from ConceptNet [10], and a list of affective words [3], and treating only four moods: *happy, sad, angry, scared*.

3. Classifying texts by the expressed emotion

We used the six basic emotions of Ekman [5]: *happiness*, *sadness*, *anger*, *disgust*, *surprise*, and *fear*. Therefore, our task was to classify into seven classes: the six basic emotions plus one class for *non-emotion*.

We used two datasets: the newspaper headlines data from SemEval 2007-Task 14, and an emotion-annotated blog corpus.

The Text Affect dataset [17] consists in newspaper headlines that were used in the SemEval 2007-Task 14. It includes a development dataset of 250 annotated headlines, and a test dataset of 1000 news headlines, for which the annotations were released after the workshop.

The annotations were made with the six basic emotions on intensity scales of [-100, 100]. The task organizers employed 6 annotators. The correlations between the score of each annotator and the average score of all the annotators ranged from 0.36 to 0.68 for different emotions.

We used all the 1250 headlines as one dataset, in order to be able to apply machine learning techniques; therefore we report results by 10-fold cross-validation on this data. We tried several classifiers for Weka [20], and SVM obtained the best results. As features we used all the 655 distinct words, as binary features (since words were not repeated in the headlines).

Table 1 presents the results of the classification for the Text Affect dataset, in the form of Precision, Recall and F-measure for each class. The total accuracy over all classed is 48%. This is rather low, but better than the baseline of 25% which classifies everything into the most frequent class. A random baseline would be even lower.

We mentioned that three teams participated in the SemEval task on this data set [17]. These methods were all unsupervised, and tested only on the development set, with very low results. More sophisticated unsupervised methods are presented

Class	Precision	Recall	F-measure
Happy (27%)	50%	50%	50%
Sad (17%)	61%	49%	55%
Fear (13%)	62%	43%	51%
Surprise (11%)	37%	19%	25%
Non-Emotion (25%)	39%	66%	49%
Disgust (2%)	80%	28%	42%
Anger (5%)	52%	16%	25%

TABLE 1. Results of the emotion classification on the Text Affect data set.

in [18]. A direct comparison is not possible because the test data was not the same, but the results are on the low side.

For our second emotion classification experiment, we used the emotion-annotated blog corpus of Aman and Szpakowicz [2]. It consists in 2090 annotated sentences (from 173 weblog posts) annotated by two judges. The inter-annotator agreement varied for different emotions. The kappa value ranged from 0.6 to 0.79.

Here are some examples from this dataset.

This was the best summer I have ever experienced. (happiness)
I dont feel like I ever have that kind of privacy where I can talk to God and cry and figure things out. (sadness)

We used several classifiers, and SVM obtained the best results. In this preliminary experiment, we used 1240 words from the sentences as binary features.

Table 2 presents the results for the emotion-annotated blog dataset, in the form of Precision, Recall and F-measure for each class, by 10-fold cross-validation. The total accuracy over all classes is 65.45%. This is higher than the baseline of 38% that classifies everything into the most-frequent class. More features from WordNet Affect and Rogets Thesaurus were used in [2], in order to obtain better results, up to an accuracy of 73.89%.

4. Classifying blogs by mood

As dataset for this task, we used the blog data set that Mishne collected for his research [11]. The corpus contains 815,494 blog posts from Livejournal, a free weblog service used by millions of people to create weblogs. In Livejournal, users are able to optionally specify their "current mood". To select their mood users can choose from a list of 132 moods, or specify additional moods. We do not use these additional moods because very few posts are annotated with them. Some statistics of this corpus are shown in Table 3. From the total posts, only 77% included an indication of the mood; we disregard the rest.

Class	Precision	Recall	F-measure
Happy (26%)	78%	68%	72%
Sad (8%)	61%	41%	49%
Fear (6%)	80%	48%	60%
Surprise (6%)	36%	39%	49%
Non-Emotion (38%)	60%	86%	70%
Disgust (8%)	70%	46%	55%
Anger (9%)	59%	36%	45%

TABLE 2. Results of the emotion classification on the blog data set.

Status	Counts
Number of Standard Moods	132
Number of User-defined Moods	54,487
Total Words	69,149,217
Average-words/Post	200
Unique Words	596,638
Individual pages	122,624
Total Weblogs	37,009
Total Posts	815,494

TABLE 3. Statistics about words and posts in the data set.

Figure 4 shows an example of blog posting in Livejournal, annotated with the mood label.

We randomly selected 144,129 blog posts as training data and 90,000 as test data. We used 132 classes, the 132 moods from which the writer can choose when he/she write the blog. We did not include in the classification additional user-defined moods, because there had very little training data (often only one instance). Figure 4 shows the most frequent moods in the test data.

As the main classifier, we used SVM from Weka. We tried other classifiers too, but the results were lower.

We have used most of features from Mishne [11], plus some additional sentiment orientation features, such as tagged words from the General Inquirer [16]. The features that we used are as follows:

(1) Frequency Counts Bag-of-Words (BoW) is the most common feature representation used in automatic text classification. We represent the words by their frequencies. We use the *Chi-Square* feature selection method to keep only the first 5000 features from the 43,109 BoW features.

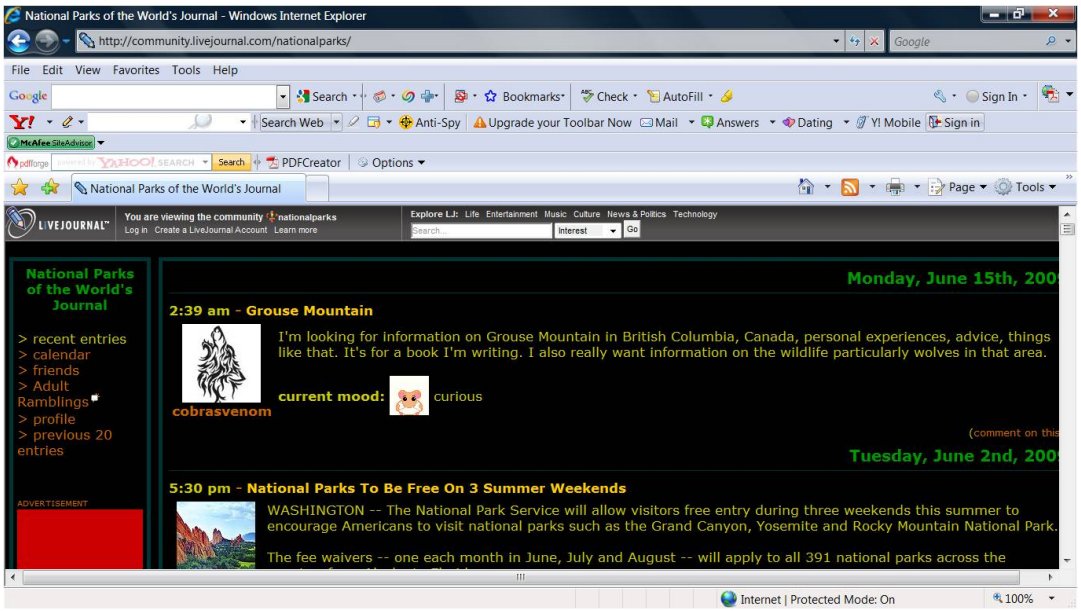

FIGURE 1. An example of blog posting in Livejournal, with the label chosen by the user.

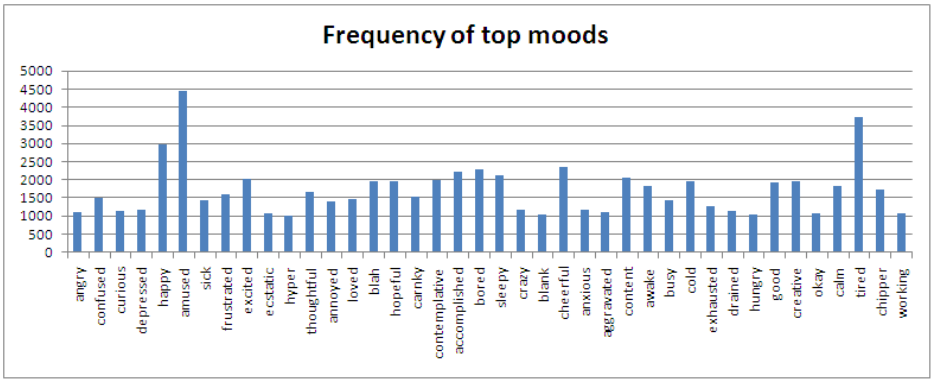

FIGURE 2. Frequency of the top moods in the test data.

(2) Length-related Features Since blog posts vary in length, we consider length features such as: the length of the document, the number of sentences, and the average number of words.

Classifier	Accuracy
Baseline (most frequent class)	7%
Flat classification (BoW)	18.29%
Flat classification (BoW+SO)	23.73%

TABLE 4. Accuracy for the initial flat mood classification.

(3) Sentiment Orientation For mood classification, the sentiment orientation (SO) of some words can be a useful feature. Several sources are predictors for sentiment orientation. We calculate six features: the total and the average orientation score for each document based on the words that are from the following resources:
   - A list of 2,291 positive words and 1,915 negative words from the General Inquirer [16].
   - A list of 21,885 verbs and nouns that were assigned a positive, negative, or neutral orientation score, Kim-Hovy list [9].
   - A list of 1,718 adjectives with their scores of polarity values, constructed by using the method of Turney and Littman [19].

(4) Special Symbols We use the special symbols called emoticons (emotional icons), that represent human emotions or attitudes. These symbols are textual representations of facial expressions, i.e. :) (smile) and ;) (wink) and so on. We used nine most popular emoticons as features.

We train a classifier into 132 mood and evaluate its performance on the Selected test set. In the result tables, we denote the features (1) and (2) under the generic name of *BoW*, and when we add features (3) and (4). We call this extended feature set *BoW+SO*. Table 4 presents the initial results for the two sets of features. The accuracy is 24.73% for *BoW+SO* and 18.29% for *BoW*. This is an improvement compared to a baseline accuracy of 7% when always choosing the most frequent class, but the accuracies are rather low.

In order to address the issue of low accuracy, we propose a hierarchical classification approach. We need to organize the closely-related classes into subclasses. Fortunately, in the Livejournal weblog service, the moods are organized in a hierarchy, shown in Figure 3. Therefore we can use this hierarchy.

For the hierarchical classification approach, we first train a classifier to classify into the 15 categories from the first level of hierarchy: *happy, sad, angry, okay, working, scared, awake, thoughtful, nerdy, indescribable, enthralled, determined, confuse, devious*, and *energetic*.

In the next step, for each node from the first level of hierarchy we extract the related instances and their mood labels. For instance, for the node *angry* we select all the documents that have the label *angry, aggravated, annoyed, bitchy, cranky, cynical, enraged, frustrated, grumpy, infuriated, irate, irritated, moody, pissed*, and

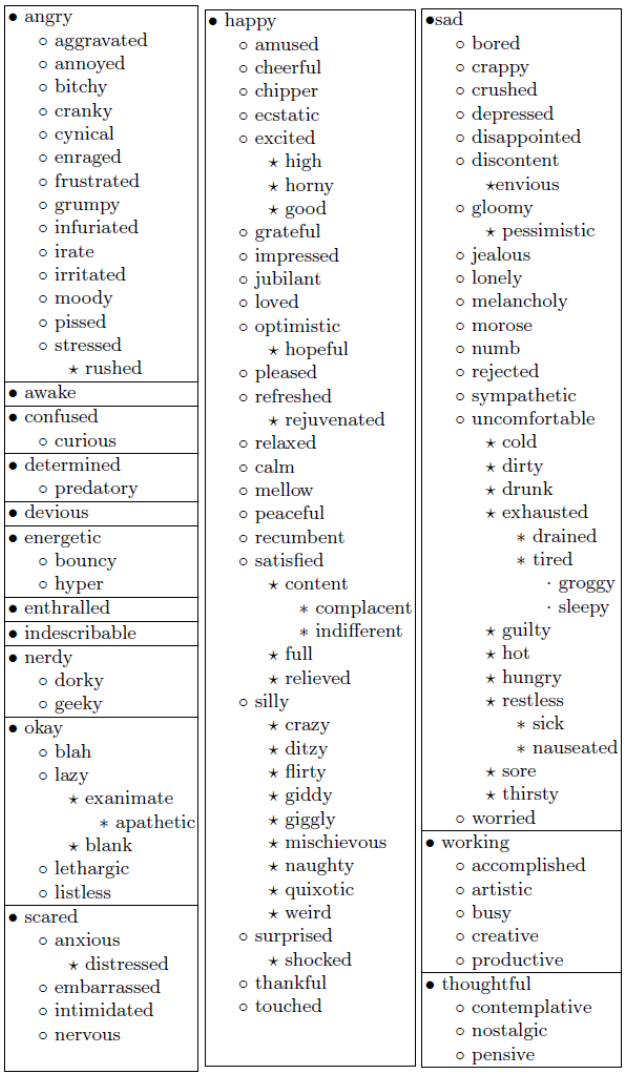

FIGURE 3. The hierarchy of the 132 moods; •: level1, ○: level2, ⋆: level3, ∗: level4, and · : level5 .

stressed. Finally, we run the classifier for each node in the second level. We repeat this procedure for each of the 15 categories from the first level of the hierarchy. We continue similar steps for the third, fourth and fifth level of the hierarchy.

	Baseline	BoW	BoW+SO
Level1	15%	40%	63.50%

TABLE 5. Accuracy for the hierarchical classification in Level 1 for both BoW and BoW+SO features.

Level2	Baseline	BoW	BoW+SO
happy	8.64%	62.72%	86.97%
sad	10.38%	66.89%	86.88%
angry	11.67%	80.13%	91.90%
okay	24.55%	78.67%	82.25%
working	25.24%	87.74%	93.29%
scared	25.97%	89.21%	95.02%
thoughtful	35.99%	91.32%	94.84%
nerdy	41.40%	90.65%	97.68%
determined	65.52%	93.25%	95.83%
confused	56.32%	85.71%	94.33%
energetic	54.05%	90.05%	96.73%
Average	32.70%	83.30%	92.33%

TABLE 6. Accuracy for classification in Level 2 for both *BoW* and *BoW+SO* features.

For the classifier that classifies into one of the 15 moods from the first level, the accuracy is 63.5% for *BoW+SO* and almost 40% for *BoW*, compared to a baseline of 15%; the results for Level1 are illustrated in Table 5.

In the next step, we have 15 classifiers in the second level, one for each node in the first level. In fact we have only 11 classifiers, because four moods did not have any children branches in the hierarchy, so for them the classification is already finished. The average accuracy was 92.33% for BoW+SO features, 83.30% for *BoW* features only, and 32.70% for a baseline of the most frequent class. The difference between the hierarchical approach with all the features and the baseline is 59.63%. There are several branches that have fewer children and show larger improvement; and there are several branches with many children that show lower performance improvement. For example the moods *happy*, *sad*, and *angry* have many children branches and the improvement is smaller. The gain in performance is bigger for moods such as *nerdy*, which has two branches. Two branches means three classes, in this case generic *nerdy* and more specific kinds of *nerdy*: *geeky* and *dorky*.

The results of the level 3 classifier are shown in Table 7. The results of Level 4 are shown in Table 8. Level 5 has only one classifier, for *tired*, with an accuracy of 96.22% for *BoW+SO* features and 87.61% for *BoW*, with a baseline of 54.44%.

Level3	Baseline	BoW	BoW+SO
uncomfortable	17.97%	71.03%	90.72%
surprised	56.18%	96.19%	97.68%
stressed	67.62%	94.58%	98.72%
silly	14.17%	74.53%	90.32%
satisfied	31.97%	80.15%	96.44%
refreshed	52.92%	95.55%	97.06%
optimistic	66.41%	90.35%	98.80%
lazy	31.37%	87.40%	96.88%
gloomy	66.21%	93.68%	99.88%
excited	35.87%	86.33%	91.75%
discontent	84.27%	92.08%	100%
anxious	58.88%	89.91%	96.85%
Average	48.65%	87.64%	95.84%

TABLE 7. Accuracy for classification in Level 3 for both BoW and BoW+SO features.

Level4	Baseline	BoW	BoW+SO
content	54.23%	89.73%	97.92%
restless	48.50%	90.27%	97.70%
exhausted	40.20%	89.17%	96.09%
exanimate	68.73%	96.46%	100%
Average	52.16%	91.40%	97.93%

TABLE 8. Accuracy for the hierarchical classification in Level 4 for both BoW and BoW+SO features.

To directly compare the results of the flat categorization to results of the hierarchical classifiers, we can cumulate the errors from all the levels. This will give a global accuracy of 55.24% for all 132 classes (for *BoW+SO*), significantly higher than 19.28% for the flat categorization. As illustrated in Table 9, the improvement in performance between the flat and the hierarchy classification is significant, especially when adding the sentiment orientation features. Our experiments and results clearly show that the hierarchical classification leads to strong performance and it is well-suited for the task. The summary of the results, shown in Table 9 clearly support above arguments.

To allow a comparison of our results to the results of Mishne [11], we run an experiment where we used only the 40 most-frequent moods. We obtain 84.89% accuracy, while Mishne obtained the best accuracy of 67%. However, the results are not directly comparable, because he used a test set, randomly chosen, with a balanced distribution of classes. Therefore we are not able to use exactly the same

Summary of the Results	
Baseline	7.00%
Flat Classification BoW	18.29%
Flat Classification BoW+SO	24.73%
Hierarchical Classification BoW	23.65%
Hierarchical Classification BoW+SO	55.24%

TABLE 9. The results of the hierarchical classification when the classifiers from all the levels are applied successively (the errors from all the levels are multiplied), compared to the results of the flat classification, for both *BoW* and *BoW+SO* features.

test set to compare our results to his result directly, but our data set is randomly chosen from the same data. The differences between our work and Mishne's work consist in the fact that we used all the 132 moods, not only the 40 most-frequent moods, and in the fact that we enhanced the feature set with more sentiment orientation features. Moreover, we use the hierarchical classification in order to improve the results.

5. Generation texts with emotions

Serious games are used for training purposes and often include exchanging messages. The player under training receives messages from various game characters, in response to his/her actions. These messages can be written manually by the game developers, or generated automatically using Natural Language Generation (NLG) techniques. For the latter approach, we proposed a template based approach with classes of variables [4]. For sentence realization we used the *SimpleNLG* package [14], which allows specifying templates in the form of Java code, and generated full sentences by adding linguistic information. The writers of the games prefer the template-based approach, but they cannot write Java code directly. Therefore we implemented an authoring tool, where they specify the components of the templates and the dependencies between them. The tool generated the Java code needed as input to *SimpleNLG*.

Since the messages are generated in the same way, and often by the same writer, they tend to sound similar. The monotony can have negative impact on the learning process, the player could get bored. There is a need to have variety in the language of different characters, in order to give the illusion of personality, including emotions and moods.

We propose to enhance the NLG process in order to automatically generate texts that are friendlier or more hostile, and texts that are more formal or more informal. When the writer implements a template, the authoring tool will allow

changing some of the words in the generated sentences, in order to achieve the impression of variety.

We collected lists of expressions that can modify sentences without changing their initial meaning, and some paraphrases. We plan to use these paraphrases as classes of variables in the authoring tool.

As a starting point, paraphrases were collected from dictionaries of synonyms. Here is one example of synonyms (or near-synonyms): *greet, accost, address, hail, salute, welcome*. Among them, *greet* is more formal, *welcome* is friendly, *accost* is hostile, etc.

Paraphrases at sentence level include various ways of expressing agreement. Here are some examples:

- Friendly agreement: *The point you made about ... is excellent. (Id like to add that ...).*
- Friendly agreement: *I agree with ..., but what about ...?*
- Hostile agreement: *Ok, whatever ...*
- Hostile disagreement: *I absolutely disagree.*

Formal language can be very polite. For example, the sentence *How can I help you?* is in contrast with the informal *What do you want?*. Another example of polite sentence is *Would you be able to write it here, please?* versus the informal *Write it here.*

We plan to extend our authoring tool [4], by adding a button for generating friendly/hostile language and one button for formal/informal language. We plan to extend this module with a verification step that rules out phrases that are ranked low by a language model for English.

6. Conclusions and Future Works

We presented supervised machine learning methods for the task of classifying text by emotions and by mood. Our future work includes improving the accuracy of the classification by emotion by adding more affect features.

We plan to use the results of our experiments in natural language generation for digital games. We need to focus on the automatic collection of paraphrases for friendly/hostile and formal/informal expressions. The evaluation of the generated texts will also be a challenge.

Acknowledgments

The authors wish to thank Soo-Min Kim and Ed Hovy for providing their polarity-tagged lists, and to Saima Aman and Stan Szpakowicz for providing the emotion-annotated blog corpus.

References

[1] C. O. Alm, D. Roth, and R. Sproat. Emotions from text: machine learning for text-based emotion prediction. In *Proceedings of the Human Language Technology Conference Conference on Empirical Methods in Natural Language Processing(HLT/EMNLP 2005)*, 2005.

[2] Saima Aman and Stan Szpakowicz. Identifying expressions of emotion in text. In *TSD*, pages 196–205, 2007.

[3] M.M Bradley and P.J. Lang. Affective norms for english words (anew). *University of Florida*, 1999.

[4] M.F. Caropreso, D. Inkpen, S. Khan, and F. Keshtkar. Visual development process for automatic generation of digital games narrative content. In *Proceedings of the Workshop on Language Generation and Summarisation (UCNLG+Sum), ACL-IJCNLP 2009*, 2009.

[5] P. Ekman. An argument for basic emotions. *Cognition and Emotion*, 6:169–200, 1992.

[6] L. Holzman and W. Pottenger. Classification of emotions in internet chat: An application of machine learning using speech phonemes. *Technical Report LU-CSE-03-002, Lehigh University*, 2003.

[7] C.E. Izard. *The Face of Emotion*. Meredith, New York, 1971.

[8] Y. Jung, H. Park, and S.H Myaeng. A hybrid mood classification approach for blog text. *LNCS*, pages 137–142, 2006.

[9] S.-M. Kim and E. Hovy. Determining the sentiment of opinions. *Proceedings of the 20th COLING*, 2004.

[10] H. Liu and P. Singh. Conceptnet, a practical commonsense reasoning tool-kit. *BT Technology Journal*, pages 211–226, 2004.

[11] G. Mishne. Experiments with mood classification in blog posts. *ACM SIGIR*, 2005.

[12] A. Neviarouskaya, H. Prendinger, and M. Ishizuka. Compositionality principle in recognition of fine-grained emotions from text. In *Proceedings of 4th International AAAI Conference on Weblogs and Social Media(ICWSM 2009)*, 2009.

[13] B. Pang and L. Lee. Opinion minding and sentiment analysis. *Foundation and Trend in Information Retrieval*, 2:1-2, 2008.

[14] E. Reiter. *SimpleNlg package, http://www.csd.abdn.ac.uk/ereiter/simplnlg*, 2007.

[15] V. Rubin, J. Stanton, and E. Liddy. Discerning emotions in texts. *AAAI-EAAT*, 2004.

[16] P. J. Stone, D. Dexter, C. Smith, S. Marshall, and D. M. Ogilvie. The general inquirer: A computer approach to content analysis. *MIT Press*, 2007.

[17] Carlo Strapparava and Rada Mihalcea. Semeval-2007 task 14: Affective text. In *SemEval 2007 Workshop*, 2007.

[18] Carlo Strapparava and Rada Mihalcea. Learning to identify emotions in text. In *SAC '08: Proceedings of the 2008 ACM Symposium on Applied Computing*, pages 1556–1560, 2008.

[19] P. Turney and M. Littman. Measuring praise and criticism: Inference of semantic orientation from association. *ACM (TOIS)*, 21(4), 2003.

[20] I.H. Witten and E. Frank. *Data Mining: Practical machine learning tools and techniques*. 2nd Edition, Morgan Kaufmann, San Francisco, 2005.

University of Ottawa, School of Information Technology and Engineering

E-mail address: diana@site.uottawa.ca, akeshtka@site.uottawa.ca, dghaz038@uottawa.ca