Classifying Public Opinion Using SentiWordnet

Charibeth K. Cheng[1], Glecer Z. Bautista[2], Michael A. S. Garcia[3], Richmond J. C. Tan[4]
College of Computer Studies, De La Salle University
Manila, Philippines
chari.cheng@delasalle.ph[1], glecer.bautista@delasalle.ph[2], mike.garcia@delasalle.ph[3], richmond_tan@dlsu.edu.ph[4]

ABSTRACT

A large amount of opinions, such as those found in blogs, fora and product reviews, are being uploaded daily as the Web continues to develop. A rich source of information may be found in these content, but they are under utilized due to difficulty in discovering and monitoring them. Existing natural language processing techniques may be tapped to organize these content making them useful for decision making. This paper focuses on the development of a system that uses text-processing techniques in organizing the sentiment of public commentaries.

Current systems are able to differentiate facts from opinions, as well as classify these opinions based on their sentiment. Clustering algorithms have been used to group related objects together. The system Vox Pop combines these three functions through lexicon-based approaches on opinion detection, sentiment classification and clustering. Opinions are differentiated from facts using a list of opinion markers. These opinions are then classified by sentiment (i.e., positive, negative or neutral) using scores produced by SentiWordNet. Comments classified under the same sentiment are further organized into groups using k-Means clustering. The detection, classification and clustering modules have accuracy rates of 50%, 50.5% and 53.85%, respectively, when compared to expert evaluations.

Keywords

Sentiment-based Clustering, Opinion Classification, Opinion Detection, Opinion Organization

1. INTRODUCTION

Sharing opinions online is now much more convenient. These opinions come in the form of reviews on consumer products, entertainment media, services, as well as personal views on almost anything. Collectively called *user-generated content*, they are found in merchant sites, blogs, discussion groups and fora. They also represent a new source of valuable information.

According to Dave Sifry [16] of Technorati, an internet search engine for blogs, an average of 50,000 blogs have been posted every hour from August 2004 to March 2006. The number of blog entries being posted online also doubles every six months. While these advancements allow convenience for opinions to have a wider reach, they also prohibit these opinions from being maximized because these opinions are not presented in an organized manner.

This research focuses on opinions found in public political commentaries. Vox Pop is one of the components of the *eParticipation* project designed for the Senate Blue Ribbon Committee [14]. *eParticipation* aims to enhance citizen participation and empowerment in two key roles of the legislature, namely, law making and executive oversight. Vox Pop is a forum-based opinion-solicitation tool, where participants contribute their opinions on a proposition posted by the moderator (presumably, the policy maker). The public is given a venue to participate in the policy-making process. Their opinions would be available for public viewing, thus, allowing access and providing opportunities to respond to opinions that may be similar or different from theirs. *eParticipation* proposes an approach for government to engage citizens in participating in policy-making; as well as empowering citizens in influencing policy-making. Anticipating numerous commentaries to be submitted, Vox Pop organizes the commentaries according to their sentiment. Organized data enable the users of the system to have more convenient access to the pulse of the public.

2. RELATED LITERATURE

2.1 Opinion Classification

Opinion detection is the identification of the presence of opinion markers in order to differentiate opinions from facts. The presence of sentiments, regardless of nature, determines whether a statement is an opinion or a fact [6]. Modal verbs, adjectives and adverbs are strong markers of opinions. Factual documents usually consist of factual sentences. On the other hand, opinionated documents are most likely made up of opinionated sentences [5].

Hovy and his colleagues [6] developed a classifier using collections of opinion-bearing and non-opinion-bearing words from WordNet, WSJ Data and Columbia Wordlist. These collections are used against models that identify opinion-bearing sentences. One model considers the total sentiment of all the words in a sentence, while another model detects the presence of even a single strong opinion-bearing word, which is enough to consider the input statement as an opinion.

Opinions may be classified by sentiment. Opinion sentiment may be classified as positive, negative or neutral, which can be identified based on the number and strength of semantically oriented words in a sentence [5] [7] [11]. Words of the same sentiment also usually occur together. Thus, a positive word is more likely to occur with other positive words than negative words.

Yu and his colleagues [5] used sentence sentiment tagging in order to detect the sentiment of an input sentence. This method measures the sentiment of a sentence through the average per word log-likelihood score, which is obtained through comparing input words with words from a seed set containing positive and negative words. Cuts-off are placed in order to determine the sentiment of an opinion. If the average score is higher than the cut-off, then the input is a positive opinion. Lower than cut-off scores define negative opinions, and scores between the cuts-off mean neutral opinions.

However, sentiments found in an opinion cannot be weighed with the use of counting only the number of positive and negative words within the sentence because of the presence of words that can shift the sentiment of the word it follows [8]. These words, called *contextual valence shifters* or CVS's, can cause the problems in determining the sentiment of a sentiment. Contextual valence shifters are classified into two groups namely: sentence-based CVS's, which include negatives, intensifiers and modal operators, and discourse-based CVS's, which are words that change the context of the sentiment like the words *but* and *however*.

Clustering is the aggregation of similar data based on a given criteria. Chiu and his colleagues [1] developed a decision support system which clusters together data of the same argument based on terms. The moderator can create a topic in which users are given the chance to have individual and independent brainstorming whether they are for, against or neutral with regards to the topic. Users can take their stand through voting and explain their views on the issue. The votes are aggregated automatically according to their stand. After the votes are properly aggregated, there will be an outlier, which consists of unique opinions of the voters. The outlier has a chance to clarify his or her stand and everybody will be allowed to revote since a particular explanation by an outlier may affect the votes of other users. The votes will be aggregated again and presented through visual representations such as a pie chart which shows the breakdown of the votes and a list of the corresponding annotations.

2.2 SentiWordNet

SentiWordNet is a lexical resource that can help in determining the sentiment of the words in the sentence. It is associated directly to each WordNet synset to determine how objective, positive or negative the words contained in the synsets are [2]. It was created by Andre Esuli and Fabrizio Sebastiani from the Consignio Nazionale delle Reserche and Universita di Padova, respectively.

SentiWordnet was built using semi-supervised synset classification. Classification relied on a set of training data that contained ternary classifiers, indicating whether a synset is Objective, Positive or Negative. SentiWordNet assigns a score for Objectivity, Positivity and Negativity in determining the sentiment of the synset. The score ranges from 0.0 to 1.0 with a sum of 1.0 for the whole synset. Hence, each synset has a sentiment of all three categories but it ranges only up to a certain degree.

SentiWordNet visualizes these final values or scores into a web interface, which is graphically represented by a triangle that each has a side for the Positive, Negative and Objective categories. The synset score is denoted by a blue circle which shows the distance of the score from the three categories in the triangular plane. The graph is easily understood as the blue circle is nearing a certain category it has a score that weighs closely to that specific category.

An example would be the word *ill(2)*, an adjective meaning *resulting in suffering or adversity*. The adjectival synset has an Objective score of 0.125, a Positive score of 0 and a Negative score of `0.875. In Figure 1, the plot of the point is near the negative side of the triangular plane. Thus, we can conclude that *ill(2)* is of negative sentiment.

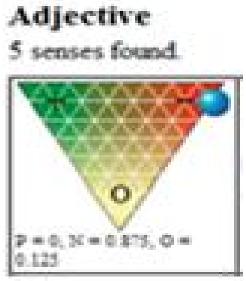

Figure 1. Graphical Visualization of *ill(2)* in the Web Interface of SentiWordNet

SentiWordNet uses prior knowledge of opinion mining as a means of weighing the synsets as Objective, Positive or Negative. Text is tagged in three steps, namely: SO-Sentiment, PN-Sentiment and Strength of the PN-Sentiment.

In SO-Sentiment, the terms within the synset is first classified whether it is Subjective or Objective. For example, the word *fish* is an objective term since it pertains to a real-world object, while the word *good* is subjective since the term does not pertain to any specific object and it needs to refer to a subject for it to have a meaning.

In PN-Sentiment, once the word is deemed as Subjective, it is further classified into two distinct classifications, Positive and Negative, since a subjective word may have a positive or a negative sense. For example, the previous word *good* gives a sentence a positive meaning as opposed to *bad*, which gives a sentence a negative meaning. Terms with similar sentiment usually have the same gloss. Mostly, terms with similar glosses have the same sentiment.

Finally, after determining the PN-Sentiment of the word, the strength of the sentiment is now weighed. The strength of the PN-Sentiment is based on the given subject matter whether an opinion is Weakly Positive, Mildly Positive or Strongly Positive [2].

3. METHODOLOGY

Vox Pop is a web-based opinion solicitation and classification system, which allows the public to voice their opinions on topics of discussion. Vox Pop detects opinions in the input and separates them from non-opinions; classifies opinions by sentiment and by topic; and presents the classified opinions in graphical representations. The system has three main modules, namely: the opinion detection module, the opinion classification module and the clustering module.

3.1 Detection

Commentaries are first gathered into a repository and then go through the first module of the system, which is the Opinion Detection Module. The first module detects quotations and opinions present within the commentary. Sentences that are tagged as opinions are forwarded to the classification module while the sentences that are untagged or tagged as quotations are disregarded in the succeeding module.

The output of the opinion detection module is the combination of the detection of quotations and the detection of opinions. These two detection processes are needed in order for the classification process to determine which parts of the commentary are to be forwarded to the Opinion Classification Module. Lines of text that are tagged as opinions are selected to undergo the classification process while sentences that are not tagged or are tagged as quotations will not undergo the classification process.

A commentary may contain a mixture of opinions and facts. Facts are sometimes presented as quotations made by other authors that the user may have used to support his stand. The module starts by detecting quotations found in the commentary. The detection of quotations is done before the actual detection of opinions to ensure that the lines of text that are tagged as quotations and are not included in the detection of opinions. The presence of quotations taken from previous posts or other passages, which can be considered as opinion spamming, could duplicate commentaries within the topic thus resulting in the presence of an additional positive or negative score. This feature prevents the occurrence of these duplicate commentaries or opinion spam for a more accurate positive or negative score. Quotation detection also prevents quoted opinions from being processed twice. Signal phrases are used to detect quotations, examples of which include *acknowledges*, *according to*, *admits*, *claims*, *said*, *confirms*, and *suggests*.

The opinion detection module uses opinion markers to detect whether a sentence is an opinion or not. Sample opinion markers include *believe*, *like*, *dislike*, *could*, *possibly*, *probably*, *often*, and *sometimes*. This prevents non-opinionated statements from being processed by the sentiment classifier.

3.2 Classification

To determine the sentiment of an opinion, it goes through four steps, namely: Part of Speech Tagging, Sentiment Score Generation, Sentiment Score Computation, and Determining the Sentiment of the Commentary. This module uses MontyTagger [9] for part-of-speech tagging and SentiWordNet [2] for sentiment score generation.

The Sentiment Score Generation submodule produces scores, which are used in computing for sentiment in the next submodule. It uses SentiWordNet as a lexical resource, which helps in determining the sentiment of the words in the sentence. SentiWordNet associates directly each WordNet synsets to three scores namely, *Obj(s)*, *Pos(s)*, *Neg(s)*, describing how objective, positive or negative the terms contained in synset are [2]. This submodule considers only the adjective and adverb synsets in computing for the sentiment score.

There are three levels of computation under this submodule, namely: Word-level, Sentence-level and Commentary-level. In the computation for the word-level sentiment, the *Positivity* and *Negativity* scores of all of the synsets of a particular adjective or adverb, depending on use, will be averaged in order to compute for the scores of that particular word.

A word ***w*** would have two sentiment scores namely, ***wordlevel_positive*** score and ***wordlevel_negative*** score, computed as follows. Table 1 explains the variables in detail.

```
wordlevel_positive(w) = total_pos / x
wordlevel_negative(w) = total_neg / x
```

Table 1. Variables used in computing for word-level sentiment

Var	Description	Equation
X	number of synsets, if the word used as an adjective or adverb	
Total _pos	total of **Pos(w_i)**, where w_i is a synset of **w** and used as an adjective or adverb	$total\_pos(w) = \sum_{i=1}^{n} Pos(w_i)$
Total _neg	total of **Neg(w_i)**, where w_i is a synset of **w** and used as an adjective or adverb	$total\_neg(w) = \sum_{i=1}^{n} Neg(w_i)$

Using the example on the word *prime*, with five synsets for adjective, the computation for the **total_pos(*prime*)** and the **total_neg(*prime*)** is shown in Table 2.

Table 2. Word-level sentiment score generation

Synset(prime)	Pos(w_i)	Neg(w_i)
prime:1	0.25	0.25
prime:2	0.0	0.0
prime:3	0.625	0.0
prime:4	0.0	0.0
prime:5	0.375	0.0
TOTAL	**total_pos** = 1.25	**total_neg** = 0.25

The word-level sentiment is then computed as:

wordlevel_positive(*prime*) = 1.25 / 5 = .25
wordlevel_negative(*prime*) = 0.25 / 5 = .05

The larger of the sentiment scores is taken as the word's sentiment, so the sentiment of the word *prime* is positive. Table 3 shows the sentiment scores and the actual sentiment of some sample words.

Table 3. Word-level sentiment of sample sentiment

	Word	POS	wordlevel_ positive	wordlevel_ negative	wordlevel _polar
Sen. 1	next	Adj	0.0	0.03125	Neg
Sen. 2	however	Adv	0.125	0.28125	Neg
	prime	Adj	0.25	0.05	Pos
	stubborn	Adj	0.0	0.6875	Neg
Sen. 3	just	Adv	0.104167	0.0	Pos
	good	Adj	0.595238	0.0	Pos
	own	Adj	0.0	0.0	Neu
	unaccoun-ted	Adj	No SentiWord Net score	No SentiWord Net score	-

As seen in Table 3, some words do not contain either positivity or negativity scores, because according to SentiWordNet, there can be objective words, such as the word *unaccounted*.

After computing for the word-level scores, the cycle would be repeated for two more times. In the first re-iteration, the Positivity and Negativity scores of all adjectives and adverbs in a particular sentence will be added and then averaged in order to come up with the scores of that particular sentence. Finally, this process is repeated one more time, this time adding and averaging the scores of sentences, in order to come up with the commentary-level scores.

After computing for the Positivity and Negativity score of the entire commentary, its sentiment would then be determined. The higher between the Positivity and Negativity scores of a commentary will serve as the Sentiment of that particular commentary.

There are two exceptions to when a commentary is classified neither Positive nor Negative. These exceptions happen when the scores are even, or when both scores are zero due to the absence of adjectives and adverbs in a commentary. In these cases, the commentary will then be classified as *neutral*.

3.3 Clustering

After being classified by sentiment, the commentaries would then be clustered by topic. Each commentary would first undergo two types of pre-processing, namely, stop words removal and stemming of words. After pre-processing the commentaries, the mean of each commentary would then be computed, and then the Euclidean distance between the commentaries and will finally be subjected to the *k*-Means Clustering proper.

The clustering algorithm used by the system is based on Collaborative Decision Support System (CoDeS) [1]. However, the implementation is slightly altered from CoDeS. While CoDeS accepts commentaries without any pre-processing for clustering, Vox Pop's clustering module accepts commentaries, which are already classified by sentiment by the classification module.

Stop words are words which bridge two ideas together and add coherence of thoughts. However, they are not unique, and thus, not important in clustering commentaries. The list of stop words used in the system is obtained from the University of Glasgow resource site [15]. Some of these words include prepositions, pronouns, articles, linking verbs and numbers. These stop words that appear in the commentary are removed since they are not important in clustering. Stop words are removed from the commentaries, which produce commentaries that contain fewer words to be clustered in the clustering proper.

According to Michael Porter [12], some words can be stripped of their affixes and still contain the same meaning. Since stemmed words arrive at words with the same meaning, these words are more useful in clustering. The system uses Porter's Stemming Algorithm [13] in order to achieve this task. This produces fewer unique words to be clustered, while increasing their frequency at the same time. This procedure lessens the possibility of classifying words with different suffixes but the same root substrings as different from each other.

The mean is the average of the individual attributes of the elements given a set of observations [10]. The mean of each word in a cluster is necessary in computing for the Euclidean Distance between commentaries and the clusters available. The mean of each word is computed by getting the number of times a word appears in a list of commentary divided by the number of commentaries in that particular cluster.

The Euclidean Distance between objects is the summation of the squares of the difference in attribute values between to observations [3]. This is computed by getting the square of the difference between the frequency of a word in a commentary, and the mean of the word on that particular commentary. This process would be iterated for each word present in the cluster, and the results are then added in order to produce the Euclidean Distance between the commentary and the cluster. The Euclidean Distance between commentaries and clusters is used in the implementation of the *k*-Means Clustering Algorithm [4]. The *k*-Means Clustering Algorithm contains three main steps, namely, the assignment step, the update step and the several iterations of the update step.

The first step in the *k*-Means Clustering Algorithm is the assignment step. This step initializes the centroids based on the number of clusters needed. The default number of clusters to be produced by the system is three. The first three commentaries will act as the initial centroids since they are the first three commentaries to enter the module. The succeeding commentaries would then be compared to both centroids in order to determine in which cluster the third commentary belong to.

After the initial run of the algorithm, all the values of centroids would then be recomputed. The new values of the centroids were derived by dividing the number of times a word appears in all of the commentaries in the cluster by the number of commentaries in a cluster. After the centroids have been recomputed, each commentary is added to the centroid which they are nearest to. This is done by getting the Euclidean Distance between a commentary and all the clusters available. The commentary is assigned to the cluster which has the smallest Euclidean Distance to it. This is called the update step. The update step is iterated until there are no more movements between the clusters.

4. RESULTS

The test corpus was built from commentaries obtained from the website of the Philippine Star Inbox World. It contains 1,002 commentaries from 22 topics. The system was evaluated by a linguist.

4.1 Detection

In order to check whether detected opinions are correctly annotated by the system, the same set of commentaries used to evaluate the classification module was fed into the opinion detection module. All 101 commentaries were split into sentences and were determined whether they are opinionated or not. 100 system-annotated opinionated and another 100 system-annotated non-opinionated sentences were randomly selected to be evaluated by the linguist. Of the 200 sentences evaluated by the linguist, one hundred sentences or **50%** matched with the detection done with the system.

The use of signal phrases/markers in quotation detection and opinion detection is not effective. First, a complete list of signal phrases is difficult to obtain. Second, the structure of the sentence must be processed to determine if a signal phrase is indeed a signal phrase. Consider the word *goes*, which is a marker of a possible quotation, such as *There's a saying that goes, "The road to hell is paved with good intentions"*. It may also be a non-signal word, as in "*There goes the alarm*".

4.2 Classification

In order to check whether classified opinions are correctly annotated by the system, 200 randomly selected classified commentaries were chosen from the test corpus. The linguist then manually identified the sentiment of each commentary. Of the 200 evaluated by the linguist, 101 commentaries or **50.5%** matched with the classification done by the system.

Four sources of errors were found. The first error is the failure to process double-negative phrases. The system gets the individual scores of the words in a commentary then adds them afterwards. An example would be the statement "*Another Aquino to dominate the Philippine as a leader has my yes. I hope Noynoy does not fail us.*" This statement is evaluated by the linguist as *positive*, as the commentary is a statement of support for Noynoy Aquino. The word *Yes* alone tells the human brain to evaluate the statement as positive. However, the formula used fails when faced with double negatives, or negative words placed next to the word *not*. The example sentence was marked as negative by the system because the words *not* and *fail*, which are both negative in nature, were evaluated separately from each other by the system. This is why the negativity score of the statement increased, instead of the positivity score increasing if it were processed as one statement *not fail*.

The second error is the presence of high sentiment words. Since the range of the scores of the words is normalized from 0 to 1, it is possible for several words of a particular sentiment to overpower a word of the opposite sentiment. An example of this would be the statement "*I believe he would make a good president of our country if given a chance, but he should be a good senator first.*" This statement is evaluated by the linguist as negative, as the *but* part of the commentary is not in support of Noynoy Aquino. However, it was marked as positive by the system. Although the word *but* has a high negativity score, the positivity score of the words *believe* and *good*, which appeared twice, overpowered the score of the word *but* because there are more words present which have high positivity scores.

The third error occurs when adjectives and adverbs are absent in a sentence. Adjectives and adverbs are opinion-bearing words. In fact, in SentiWordNet, adjectives and adverbs have non-zero positivity and negativity scores. However, opinionated statements may not contain adjectives and adverbs, such as "*I agree, one hundred percent, yes*". This statement is evaluated by the linguist as positive, as the word *yes* is enough to tell that the sentiment is positive. However, it was marked by the system as neutral because the words *a*, *percent* and *yes* are nouns, while the word *hundred* is an adjective, but has zero positivity and negativity scores. Although many opinion words are adjectives and adverbs, nouns (*garbage*, *trash*, *crap*) and verbs (*hate*, *love*, *like*, *dislike*) can also indicate opinions.

Fourth, processing should not be limited to individual words, because there are opinion phrases or idioms such as "*full of crap*" and "*cost an arm and a leg*".

4.3 Clustering

In order to check whether the clusters produced by the system are correct, two topics containing 81 commentaries were chosen to be clustered by the system and evaluated by the linguist afterwards. In generating the clusters, the commentaries in each topic were first segregated into three clusters, produced by the opinion classification module of the system. Afterwards, each sentiment-segregated cluster was further segregated into three smaller clusters. Thus, all in all, eighteen clusters were produced by the system for the evaluation of the linguist. The clusters generated by the system were analyzed by the linguist whether 1) the commentaries in each cluster are related with each other, and 2) why some commentaries are singled out into single clusters. Of these eighteen clusters, thirteen contain multiple commentaries, while the remaining five contain only single commentaries.

In the first topic, three unrelated clusters were deemed as such because the views in them are complex, vary with each other and some go beyond the intended topic. In the second topic, the three unrelated clusters were deemed as such because their views vary from each other. Another unrelated cluster contained an opinion and a declaration of support. These commentaries are grouped together because of the similar words that are found within them, such as *Filipino* and *honesty* in the first topic. However, the clustering algorithm does not take into account synonyms or words that are similar in context, resulting to some clusters being mismatched. All in all, the linguist evaluation shows a **53.85%** accuracy of the clustering module.

In the process of clustering, five commentaries were not clustered and instead, were isolated from the other commentaries. Of the five clusters containing only one commentary each, two of them were evaluated by the linguist being isolated because they sound like factual statements. On the other hand, the other two were evaluated by the linguist as being isolated because they contain alternate reasons on why they agree or disagree with the topic. Finally, the last cluster containing only one commentary was probably isolated because "it cites very specific instances", as the linguist points out.

These commentaries were isolated probably because the default number of clusters (three) is not the optimum number of clusters for the set of commentaries. An example would be the positive clusters under the second topic. When the number of clusters was set to three, one commentary was isolated while the other two clusters contained three and thirteen commentaries respectively. However, when the number of clusters was set to two, no more commentaries were isolated and two clusters containing multiple commentaries are formed.

5. CONCLUSION

The study has focused on the development of a system that uses text processing techniques in organizing the sentiments of public commentary.

The opinion detection module includes the detection of quotations and opinions given input commentaries. The study shows that quotation detection is important in this kind of system since it can greatly affect the next module, which is the opinion classification module. Quotation detection prevents quotations from being classified by the next module, thus providing more accurate results for classification. However, the study also shows that having opinion detection in this kind of system is not that important since the user inputs of the system are most likely opinionated in nature.

The opinion classification module includes part-of speech tagging, sentiment score generation via SentiWordNet and word, sentence and commentary-level score computations. The study shows that part of speech tagging is important, because adjectives and adverbs do have the linguistic basis in classifying commentaries by sentiment. However, the study also shows that SentiWordNet should not be the sole tool used in dealing with sentiment, as it only produces the score of a word, and it does not consider more complex factors such as double negatives and idioms. Nouns and verbs can also indicate opinion.

The clustering module includes stop words removal, stemming and the use of the *k*-Means clustering algorithm. The study shows that pre-processing techniques such as stop words removal and stemming are necessary in clustering as they filter commentaries, preventing non-relevant words such as prepositions, articles and pronouns from being used as the basis for clustering. However, the study also shows that having a fixed number of clusters, generated by the *k*-Means clustering algorithm, which is three in this case, is not the most optimal solution for all cases. If there are only few commentaries to be clustered, setting the number of clusters to a smaller number such as two might be more optimal. Conversely, three clusters might not be sufficient for a larger dataset, such as the ones containing thousands of commentaries in them.

6. ACKNOWLEDGMENTS

This project is supported by De La Salle University, and the International Development Research of Canada

7. REFERENCES

[1] Chiu, C., Dy, J., Lee, P. & Liu, M. (2008). *Collaborative Decision Support System*. Undergraduate Thesis, College of Computer Studies, De La Salle University, Manila, Philippines.

[2] Esuli, A. & Sebastiani, F. (2006). SENTIWORDNET: A Publicly Available Lexical Resource for Opinion Mining. In *Proceedings of the 5th Conference on Language Resources and Evaluation* (LREC06), 417-422, Genova, Italy, May 24-26, 2006.

[3] Fung, G. (2001). *A Comprehensive Overview of Basic Clustering Algorithms*. *Technical Report*, University of Wisconsin, Madison, Wisconsin, USA.

[4] Hartigan, J. (1975). Clustering Algorithms. *Wiley Series in Probability and Mathematical Statistics*. John Wiley & Sons, New York-London-Sydney, Vol. xiii, pp. 351.

[5] Hatzivassiloglou, V. & Yu, H. (2003). Towards Answering Opinion Questions: Separating Facts from Opinions and Identifying the Sentiment of Opinion Sentences. In *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing* (ACL03), 129-136, Sapporo, Japan, July 11-12, 2003.

[6] Hovy, E. & Kim, S.M. (2005). Automatic Detection of Opinion Bearing Words and Sentences. In *Proceedings of the Second International Joint Conference on Natural Language Processing* (IJCNLP05)*: Companion Volume*, 61-66, Jeju Island, Republic of Korea, October 11-13, 2005.

[7] Hovy, E. & Kim, S-M. (2004). Determining the Sentiment of Opinions. In *Proceedings of the 20th International Conference on Computational Linguistics* (COLING04), 1367-1373, University of Geneva, Switzerland, August 23-27, 2004.

[8] Huang, C. & Li, S. (2009). Sentiment Classification Considering Negation and Contrast Transition. In *Proceedings of the 23rd Pacific Asia Conference on Language, Information and Computation* (PACLIC 2009), 307-316, Hong Kong, December 3-5, 2009.

[9] Liu, B. (2010). Sentiment Analysis and Subjectivity. *Handbook of Natural Language Processing*, Second Edition, 2010.

[10] Murphy, R. (2000). Multivariate Distance and Similarity. *Cytometry Development Workshop*.

[11] Okumura, M., Suzuki, Y. & Takamura, H. (2006). Application of Semi-supervised Learning to Evaluative Expression Classification. In *Proceedings of the 7th International Conference on Intelligent Text Processing and Computational Linguistics* (CICLing 2006), 502-313, Mexico City, Mexico, February 19-25, 2006.

[12] Porter, M. (1980). An Algorithm for Suffix Stripping, *Program*, 14(13):130-137.

[13] Porter, M. (2006). *The Porter Stemming Algorithm*. Retrieved February 5, 2009, from http://tartarus.org/~martin/PorterStemmer/

[14] Roxas. R., Borra, A., Cheng, C. & Ona, S. (2010). *Developing Natural Language Processing / Data Mining Application for e-Legislation*. Interim Technical Report - IDRC Project Number: 104935.

[15] Sanderson, M. (n.d.). *Stop Word List*. Retrieved July 28, 2009, from Department of Computer Science, University of Glasgow: http://www.dcs.gla.ac.uk/idom/ir_resources/linguistic_utils/stop_words

[16] Sifry, D. (2006). *State of the Blogosphere*, August 2006. Retrieved November 2009, from http://www.sifry.com/alerts/archives/000436.html