An Analytical Approach to Assess Sentiment of Text

Mostafa Al Masum Shaikh[*], Helmut Prendinger[†], and Mitsuru Ishizuka [*]
[*]Department of Information and Communication Engineering, Tokyo, Japan,
email: *mostafa@mi.ci.i.u-tokyo.ac.jp, ishizuka@i.u-tokyo.ac.jp*
[†] Digital Contents and Media Sciences Research Division, Tokyo, Japan, email: *helmut@nii.ac.jp*

***Abstract*— Sentiment (i.e., bad or good opinion) described in texts has been studied widely, and at three different levels: word, sentence, and document level. This paper describes a well-founded approach for the task of sentence level sentiment analysis by studying the relationship between sentiments conveyed through texts and structure of natural language by a method of numerical analysis. Different approaches have been employed to "sense" sentiment, especially from the texts, but none of those ever considered the valence based appraisal structure of sentiments which we have employed. Therefore the paper describes an approach to sense sentiments contained in a sentence by applying a numerical-valence based analysis. To meet this objective a linguistic tool, SenseNet, has been developed that provides lexical-units on the basis of each semantic verb frame obtained from the input sentence; assigns a numerical value to those based on their sense affinity; assesses the values using rules; and finally outputs sense-valence for each input sentence. Several experiments with a variety of datasets containing data from different domains have been conducted. The obtained results indicate significant performance gains over existing state-of-the-art approaches.**

***Keywords*—Sentiment Analysis, Affect Sensing from Text**

I. Introduction and Background

It is noticed that all the previous approaches for analyzing texts for sensing sentiments have commonly employed techniques like, keyword spotting [29]; lexical affinity [27]; statistical methods [14]; pre-processed models; a dictionary of affective concepts and lexicon [18]; commonsense knowledgebase [8][9]; fuzzy logic [25]; knowledge-base approach [5], machine learning [20]; and domain specific classification [11]. The latest attempt, e.g. [8], have categorized texts into a number of emotion groups such as the six so-called "basic" emotion based on "facial expression variables" proposed by Ekman, which we believe are not adequate for classifying emotions expressed by textual information for at least one of the reasons like not considering the cognitive structure of emotions; or not considering adequate emotion types for classification; or not considering a Belief-Desire-Intention (BDI) model of the user (for criticism see [21],[23]). According to a linguistic survey [14], across all of the studies described there, only 4% were emotional (adjectives) words used in written texts. This indicates that evaluating the affective lexicons is not sufficient to recognize affective information from texts and raises the suspicion that methods like machine learning or keyword spotting or lexical affinity might not perform well for this objective and these are strongly been criticized in [8]. Statistical methods are well suited for psycholinguistic analysis for inferring peoples' attitude, social-class, standards etc.[14]. Fuzzy logic assesses input text by spotting regular verbs and adjectives, without their semantic relationships, that have pre-assigned affective category, centrality and intensity. Like machine learning, it cannot produce a legitimate analysis for smaller text units such as sentences. Knowledge-base approach studied how humans express emotions in face to face communication and based on this study, two-dimensional (pleasant/unpleasant, active/passive) affective lexicon database and a set of rules that describes dependencies between linguistic contents and emotions have been developed. In our opinion this approach is almost similar to keyword-spotting and therefore not suitable for sentence-level sentiment recognition. The use of domain specific corpora for emotional classification of text has shown very promising results regarding sentiment analysis of blogs, but it requires special tuning on data necessary to build category specific classifiers for human interest domains. A detail review about the methods to classify news based on sentiment is been discussed in [22],[23]. Of the techniques so far we have encountered, none of those ever considered the appraisal structure of natural language for sentiment analysis.

II. Our Approach

The interpretation of opinions is usually debatable affair even for humans. However our approach is an attempt towards this task. We have employed the following linguistic tools to make our system functional.

A. Contextual-Valence Assessment

Before we explain the algorithm, we first discuss its underlying data structure.

1) Input. The smallest input to the system is a sentence S. A paragraph P, containing one or more sentences can also be processed by the system.

2) Processing elements. We assume the input is a Paragraph *P*, containing *n* sentences, such that $P = \{S_1, S_2,...,S_i,..., S_n\}$ and $1 \leq i \leq n$. As a sentence S_i may have one or more verbs, the semantic parser may output one or more triplet(s) for S_i. We represent S_i as a set of *m* triplets *T*, i.e., $S_i = \{T_1, T_2,...T_j,..., T_m\}$, whereby $1 \leq j \leq m$. A triplet T_j has the following form: ⟨actor, action, concept⟩. The triplet elements actor and concept have the following form, ⟨name, type, attribute⟩. The action has the form ⟨name, status, attribute⟩. An attribute is either an empty set or non-empty set of words. For example, the input *S*, 'The President called the space shuttle Discovery on Tuesday to wish the astronauts well, congratulate them on their space walks and invite them to the White House.', the following four triplets are obtained for the four verbs.

T_1 = ⟨⟨President, Concept, {the}⟩, ⟨call, past, {time: Tuesday, dependency: to}⟩, ⟨discovery, Named-Entity, {the, space, shuttle}⟩⟩

T_2 = ⟨⟨President, Concept, {the}⟩, ⟨wish, infinitive, {dependency: and}⟩, ⟨astronaut, Concept, {the, adv: well}⟩⟩

T_3 = ⟨⟨President, Concept, {the}⟩, ⟨congratulate, infinitive, {dependency: and}⟩, ⟨astronaut, Concept, {goal: space walk}⟩⟩

T_4 = ⟨⟨President, Concept, {the}⟩, ⟨invite, infinitive⟩, ⟨astronaut, Concept, {place: white house}⟩⟩

3) Knowledgebase. The knowledge-base of the system has been discussed in Section 3.4. Using that data source, the system builds the following computational data-structure that is consulted to process the input text. The verbs are classified into two groups, affective verb (AV) and non-affective verb (V) group. The verbs having the tag <affect> in the knowledge-base are members of AV. Both AV and V are further partitioned into positive (AV_{pos}, V_{pos}) and negative (AV_{neg}, V_{neg}) groups on the basis of their prior valences. Similarly, adjectives (ADJ), adverbs (ADV), concepts (CON) also have positive and negative groups indicated by ADJ_{pos}, ADJ_{neg}; ADV_{pos}, ADV_{neg}; and CON_{pos}, CON_{neg}; respectively. For a named entity (NE) the system creates three kinds of lists, namely ambiguous named entity (NE_{ambi}), positive named entity (NE_{pos}) and negative named entity (NE_{neg}). The named entity that has a different sign for the valence of 'genre' and 'general sentiment' fields is a member of NE_{ambi}.

4) Algorithm. The core algorithm underlying our system can be summarized as follows.

Input: $P=\{S_1, S_2, ... S_n\}$ // a Paragraph which is a set of sentences

Output: $V = \{V_1, V_2, ... V_n\}$ // indicates valence for each sentence

Pseudo Code for Processing:

```
Begin
for each $S_i$ in $P$ do //assume $1 \le i \le n$
  tripletSet$_i$ = getSemanticParsing ($S_i$)
          //the output of Semantic Parser is a set of Triplets for each sentence.
  for each triplet $T_j$, in tripletSet$_i$ do
    //we assume $1 \le j \le m$, m triplets
     actorValence = ContextualValenceAttrib (actorPriorValence, actorAttributes)
     actionValence =ContextualValenceAttrib (actionPriorValence,  actionAttributes)
     objectValence = ContextualValenceAttrib (objectPriorValence, objectAttributes)
     actionObjectPairValence=setActionObjectPairVal (actionValence, objectValence)
     tripletValence = setTripletValence (actorValence, actionObjectPairValence)
     tripletValence = handleNegationAndConditionality (tripletValence, $T_j$)
     tripletDependency = if the token "dependency" is found then 'true' else 'false'
  tripletDependencyType = if the token "dependency: to" is found then assign 'to_dependency' else assign 'not_to_dependency'
  tripletResult$_j$ = {tripletValence, tripletDependency, tripletDependencyType}
 loop until all triplets are processed
  contextualValence = processTripletLevelContextualValence (tripletSet$_i$)
  m = sizeof(contextualValence)
  sentimentScore = $\text{average}(\sum_{k=1}^{m} \text{abs}(\text{contextualValence}_k))$
  valenceSign = getResultantValenceSign (contextualValence)
  SentenceValence$_i$ = sentimentScore * valenceSign
loop until all sentences are processed
End
```

Here are some example rules to compute contextual valence using attributes (e.g., adjectives and adverbs).

- ADJ_{pos}+ (CON_{neg} or NE_{neg})→ neg. Valence (e.g., strong cyclone; nuclear weapon)
- ADJ_{pos}+ (CON_{pos} or NE_{pos})→ pos. Valence (e.g., brand new car; final exam)
- ADJ_{neg} + (CON_{pos} or NE_{pos})→ neg. Valence (e.g., broken computer; terrorist gang)
- ADJ_{neg} + (CON_{neg} or NE_{neg})→ neg. Valence (e.g., ugly witch; scary night)

So we notice that the sign of the valence is toggled by the adjectives when there is a negative scored adjective qualifying a CON_{pos} or NE_{pos}. In other cases the sign of respective CON or NE is unchanged. The resultant valence (i.e., actor valence or object valence) is also intensified than the input CON or NE due to ADJ.

For adverbs the following rules are applied. We have some adverbs tagged as <except> to indicate exceptional adverbs (e.g., hardly, rarely, seldom etc.) in the list. For these exceptional adverbs we have to deal with ambiguity as explained below.

- ADV_{pos} + (AV_{pos} or V_{pos})→ pos. Valence (e.g., write nicely; sleep well)
- ADV_{pos} + (AV_{neg} or V_{neg})→ neg. Valence (e.g., often miss; always fail)
- ADV_{neg} + (AV_{pos} or V_{pos})→ neg. Valence (e.g., rarely complete; hardly make)
- ADV_{neg} + AV_{pos} → pos. Valence (e.g., badly like; love blindly)
- ADV_{neg} + (AV_{neg} or V_{neg})→ ambiguous (e.g., hardly miss; kill brutally)

Hence, the rules to resolve the ambiguity are:

- ADV_{neg} (except) + (AV_{neg} or V_{neg})→ pos. Valence (e.g., rarely forget; hardly hate)
- ADV_{neg} (not except)+(AV_{neg} or V_{neg})→ neg. Valence (e.g., suffer badly; be painful)

The contextual valence of Action-Object pairs is computed based on the following rules taking the contextual valence of action and object into consideration.

- Neg. Action Valence + Pos. Object Valence → Neg. Action-Object Pair Valence (e.g., kill innocent people, miss morning lecture, fail the final examination, etc.)

- Neg. Action Valence + Pos. Object Valence → Pos. Action-Object Pair Valence (e.g., quit smoking, hang a clock on the wall, hate the corruption, etc.)
- Pos. Action Valence + Pos. Object Valence → Pos. Action-Object Pair Valence (e.g., buy a brand new car, listen to the teacher, look after you family, etc.)
- Pos. Action Valence + Neg. Object Valence → Neg. Action-Object Pair Valence (e.g., buy a gun, patronize a famous terrorist gang, make nuclear weapons, etc.)

We are aware that the above rules are naive and there are exceptions to the rules. In the sentences "I like romantic movies" and "She likes horror movies" the rules fail to detect both as conveying positive sentiment because "romantic movies" and "horror movies" are considered positive and negative, respectively. In order to deal with such cases we have a list of affective verbs (AV_{pos}, AV_{neg}) that uses the following rules to assign contextual valence for an affective verb.

- AV_{pos} + (pos. or neg. Object Valence) = pos. Action-Object Pair Valence (e.g., I like romantic movies. She likes horror movies.)
- AV_{neg} + (neg. or pos. Object Valence) = neg. Action-Object Pair Valence (e.g., I dislike digital camera. I dislike this broken camera.)

The rules for computing valence of a triplet are as follows. Pronouns (e.g. I, he, she etc.) and proper names (not found in the listed named-entity) are considered as positive valenced actors with a score 1 out of 5 for simplicity. The rules are:

- (CON_{pos} or NE_{pos})+ Pos. Action-Object Pair Valence→ Pos. Triplet Valence (e.g., the professor explained the idea to his students.)
- (CON_{pos} or NE_{pos}) + Neg. Action-Object Pair Valence→ Neg. Triplet Valence (e.g., John rarely attends the morning lectures.)
- (CON_{neg} or NE_{neg}) + Pos. Action-Object Pair Valence→ Tagged Negative Triplet Valence (e.g., the robber appeared in the broad day light.) to process further.
- (CON_{neg} or NE_{neg}) + Neg. Action-Object Pair Valence→ Neg. Triplet Valence (e.g., the strong cyclone toppled the whole city.)

For example, the input sentence "The robber arrived with a car and mugged the store-keeper." outputs two 'tagged negative triplet valence' values for the actor (robber) where the 'action-object pair valence' for ["arrive, car"] and ["mug, store-keeper"] are positive and negative, respectively. For such cases where a negative valenced actor is associated with at least one 'negative action-object pair valence', the tagged output is marked with a highly negative valence.

But if a negative valenced actor is associated with all positively scored 'action-object pair valence' the 'tagged negative triplet valence' is toggled to positive. For example, "The kidnapper freed the hostages and retuned the money." gives two tagged negatives scores (i.e.; -8.583 and -9.469) for two positive "action-object pair valence" (i.e., ['free, hostage'] and ["return", "money"]). Hence, the system finally assigns a positive valence because the negative valenced actor is not associated with any negative 'action-object pair valence". This implies that an action done by a negative-role actor is not necessarily always negative. We also consider the cases of negation and conditionality as discussed in [6],[27].

B. Sentiment Assessment

In the previous subsection we described how valence is assigned to a 'Triplet'. Now we explain how sentiment is assessed for a sentence (i.e., assessing contextual valence for Triplets). The data structure 'tripletResult' has a field named 'tripletDependency' to indicate inter-dependency between two Triplets. If interrelated Triplets are found, the function *setContextualValence()* is invoked to set the contextual valence for those triplets. The algorithm of this function is described below:

Semantic parser may output a dependency tag like, "dependency: to", "dependency: and", "dependency: but", "dependency: nonetheless", "dependency: as" etc. associated with a triplet depending on the presence of connectives or conjunctions in an input sentence. In order to deal with triplets having such "dependency" tag, we considered all the dependency relations into two types namely, "to_dependency" (i.e., "dependency: to") and "not_to_dependency" (i.e., all others except "dependency: to") and applied different sets of rules to calculate a contextual valence for these two dependency types.

If there are two triplets, T_1 and T_2 where T_1 has a "to_dependency" relationship with T_2, we have,
|contextualValence|= (|valence of T_1| + |valence of T_2|) / 2

- Pos. valence of T_1 + Pos. valence of T_2 → Pos. contextualValence (e.g., I am interested to go for a movie.)
- Neg. valence of T_1 + Pos. valence of T_2 → Neg. contextualValence (e.g., It was really hard to swim across this lake.)
- Pos. valence of T_1 + Neg. valence of T_2 → Neg. contextualValence (e.g., It is easy to catch a cold at this weather.)
- Neg. valence of T_1 + Neg. valence of T_2 → Pos. contextualValence (e.g., It is difficult to take bad photo with this camera.)

The rules for other than "to_dependency" relationship are:
|contextualValence|=(|(valence of T_1)+(valence of T_2)|) /2

- Pos. valence of T_1 + Pos. valence of T_2 → Pos. contextualValence (e.g., they got married and lived happily.)
- Neg. valence of T_1 + Pos. valence of T_2 → Pos. contextualValence (e.g., John was not a regular student but he finally scored good grades.)
- Pos. valence of T_1 + Neg. valence of T_2 → Neg. contextualValence (e.g., the movie was very interesting but at the end it became monotonous.)
- Neg. valence of T_1 + Neg. valence of T_2 → Neg. contextualValence (e.g., I feel very sad when my paper gets rejected.)

The function *processTripletLevelContextualValence()* is explained in section 3.5. The function processes all the triplets and assigns the contextual valence depending on their relationships by applying the aforementioned rules. The algorithm to process the triplets of a sentence is stated below.

```
function processTripletLevelContextualValence
($tripletSet_i$)
Begin
M= sizeOf($tripletSet_i$)
ContextualValence =[ ]
for k = 1 to M-1 do
    $R_1$ := $tripletResult_k$
    $R_2$ := $tripletResult_{k+1}$
    if $R_1$.tripletDependency = true and
$R_1$.tripletDependency != "to_dependency"

$ContextualValence_k$=setContextualValence($R_1$.tripletValence,$R_2$.tripletValence, "not_to_dependency")
    else if  $R_1$.tripletDependency=false
        $ContextualValence_k$ = $R_1$.tripletValence
end loop k
for k = 1 to M-1 do
    $R_1$ := $tripletResult_k$
    $R_2$ := $tripletResult_{k+1}$
    if $R_1$.tripletDependency = true and
$R_1$.tripletDependencyType = "to_dependency"
        if $ContextValence_{k+1}$ != null then
        Begin
          $ContextualValence_k$ =
setContextualValence($R_1$.tripletValence,
$ContextValence_{k+1}$, "to_dependency")
          $ContextualValence_{k+1}$= null
       End
   Else
$ContextualValence_k$ =
setContextualValence($R_1$.tripletValence,
$R_2$.tripletValence, "to_dependency")
end loop k
return ContextualValence
End
```

The average of the absolute values of the list 'contextualValence' is assigned as the 'sentimentScore' for a sentence, S. The 'valenceSign' is set +1 if the count of positive tripletValence is greater than the number of negative ones and vice versa. If both negative and positive counts are equal then either +1 or -1 is set according to the sign of the value of the list whichever is the maximum considering their absolute values. The value of 'sentimentScore' is multiplied with 'valenceSign' to get 'sentenceValence' and it is the valence we get for a sentence. According to the scoring system the range of 'sentenceValence' is ±15 since the maximum and minimum valence of a triplet can be 15 and -15 respectively. The detail implementation of SenseNet has been discussed in [21],[23].

III. Test and Evaluation

We intend to evaluate our system at the sentence level emotion sensing. To this end, we performed system evaluation in two ways, first, by comparison with a "gold standard", and second, by comparison to another state-of-the-art system (Liu et al. 2003).

A. *The Datasets*

We use four datasets to test our method of sentiment assessment for sentence level. The evaluation to assess the accuracy of sentence level sentiment recognition is performed by comparing system results to human-ranked scores (as "gold standard") for two datasets.

The first one, Dataset A, is created by collecting 200 sentences from internet based sources for reviews of products, movies, and news (My Yahoo! 2006), and email correspondences. It was scored by 20 human judges according to positive, negative, and neutral sentiment affinity by an online survey (the detail can be found here, http://ita.co.jp/research/survey/ [one can login using a guest username]). The judges were instructed to login to the online survey system to read the sentences and score each sentence in terms of 'Sentiment' (i.e., negative, positive or neutral) and "Intensity" (i.e., low, mid, high, extreme) of sentiment by selecting radio buttons. After the survey the number of positive, negative, and neutral sentences has been decided according to the scores for which maximum number of judges are found unanimous for each sentence. For example, the input sentence *"She is extremely generous, but not very tolerant with people who don't agree with her."*, was rated as negative by 14 judges (out of 20), as neutral by five judges, and as positive by one judge. Since the majority of the judges voted this sentence as a negative sentence, the sentence is considered as a negative sentence in our "gold standard" dataset. The inter-rater agreement was calculated using Fleiss' Kappa statistics. The Kappa coefficient (κ) for sentence scoring is 0.782, showing good reliability of inter-rater agreement. This dataset contains 90 positive, 87 negative, and 23 neutral sentences. More detail about the dataset is given in Table I.

The second dataset, Dataset B, is the sentence polarity dataset v1.0 (i.e., introduced in Pang and Lee at ACL 2005. at http://www.cs.cornell.edu/People/pabo/movie-review-data/) introduced in Pang and Lee (2005). The dataset contains 5331 positive and 5331 negative classified sentences or snippets (i.e., only the subjective opinion sentences of movie reviews). The primary motivation of using these two datasets is that they contain individual sentences classified as positive, negative or neutral (for Dataset A), or positive or negative (Dataset B), which is accord with the purpose of our first experiment, namely, to answer how efficiently the system can assess sentiments at sentence level.

TABLE I. Input datasets

Dataset	Data Attributes	Data Source
Dataset A	Data collected from various domains. 90 Positive, 87 Negative and 23 Neutral sentences. More specifically the contexts and sentences are: Email: 6 pos, 5 neg, & 2 neu Product Review: 21 pos, 21 neg, 6 neu Movie Review: 15 pos, 16 neg, 5 neu News: 48 pos, 45 neg, 10 neu	Authors managed to collect the data and scoring is done by an online survey.
Dataset B	Collected from Movie Review (Rotten Tomatoes pages). There are two files. One contains 5331 positive snippets and other has 5331 negative snippets. Each line in these two files corresponds to a	Sentence polarity dataset v1.0. Introduced

	single snippet (usually containing roughly one single sentence); all snippets are down-cased. The Data source is here: http://www.cs.cornell.edu/People/pabo/movie-review-data/	in Pang and Lee at ACL 2005.

B. *Comparing to Gold Standard*

In our first experiment, since Dataset A has neutral sentences, the system performs as a three-class (i.e., positive, negative and neutral) classifier. Hence, we set different valence ranges to signal the neutrality of sentiment. The motivation is to identify the valence range for which the system shows the highest F-score in terms of classifying negative, positive or neutral sentiment bearing sentences with respect to the gold standard. The details of the experimental result are given in Appendix B. According to the result, increasing the neutral range increases the recall of neutral sentences, but decreases recall for positive and negative sentence classes. We noticed that after a certain range (here, -6 to 6), the recall for the neutral sentence class is maximized (100%), and the recall for other two classes becomes lower than 80% for the range -4.5 to 4.5. We also calculated the average of recall, precision, and F-score of the three classes for each neutral range and plotted it in line graphs, as shown in Fig. 4. According to Fig. 1, the system achieves the highest accuracy (84%) for the ranges ±0.5 and ±1.0, but it shows the highest average recall (81.04%), precision (76.49%) and F-score (78%) for the neutral range ±3.5. Since the highest F-score is achieved at this point, we decided this valence range to classify a sentence as 'neutral', i.e. the 'sentenceValence' score resides within this range.

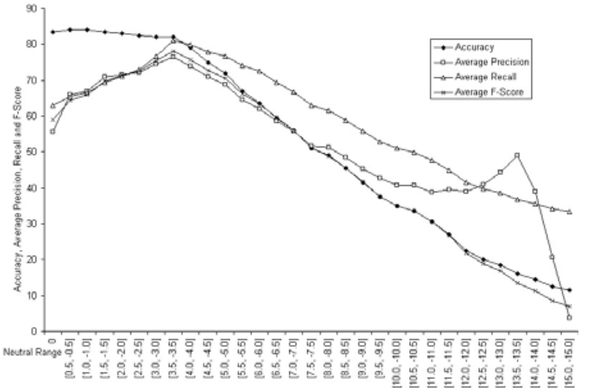

Fig. 1. Relationship between the 'Neutral Range' of the system to signal neutrality of a sentence and other system performance measures namely, Accuracy, Average Precision, Recall, and F-Score for three classes.

C. *Comparing to Gold Standard and SVM based approaches*

Dataset B has only two types of sentences, either positive or negative. Hence, in this experiment our system acts as a two-class (i.e., positive, negative) classifier. We compared the performance of our system with several other methods. Table II summarizes the accuracy of different approaches including ours for this dataset.

TABLE II. ACCURACY RESULTS OBTAINED FOR DATASET B USING DIFFERENT APPROACHES.

Approaches	Accuracy
Unigram SVM	75.11%
Bi-gram SVM	71.04%
Linguistic Tree Transform SVM	84.09%
Our Approach	91.53%

From this database, the first 4000 sentences were used to form a training set, and the remaining 1331 sentences were used to test accuracy performance using SVM approaches according to the experiments regarding SVM described in [3] and [13]. The lists output from the Linguistic Tree Transformation Algorithm were arranged into frequency SVM model form (with the SVM-light software package). Performance was tested against a frequency unigram SVM model and a frequency bi-gram SVM mode. In our experiment, 10,662 sentences were input to the system and obtained a recall of 90.62% and 92.44%, with precision of 92.07% and 91.01%, for classifying positive and negative sentences, respectively.

D. *Comparison to the EmpathyBuddy System of Liu et al. [8]*

Although the system *EmpathyBuddy* [8] does not directly assess sentiment of text (as our system does), it is known for its outstanding performance in analyzing emotion from text of smaller input size (e.g. a sentence). Like our system, Liu's system is a rule based system. It is said to be the best performing system for sentence-level emotion sensing. On the practical side, it is freely available on the internet, and thus easily available for comparison.

In order to compare the output of Liu's system to our scoring model, we considered 'fearful', 'sad', 'angry', and 'disgust' emotions as belonging to the negative sentiments, and 'happy' and 'surprise' as belonging to the positive sentiments. These are the emotions that *EmpathyBuddy* can recognize. The system considers 'surprise' as a positive emotion, and hence it resolves one of the example sentences mentioned in Liu et al. [8], *"it's a gorgeous new sports car!"* as a positive one, which as the "surprise" emotion associated to it. For each sentence a vector containing the percentage value afferent to each emotion is returned by this system. For example, for the two sentences *"It is difficult to take bad photo with this camera."*, and *"Of all my relatives, I like my aunt Martha the best."*, *EmpathyBuddy* outputs the following sets of emotions along with their level of percentage: {surprised (67%), angry (38%), sad (31%), happy (0%), fearful (0%), disgusted (0%)} and { fearful (20%), happy (0%), sad (0%), angry (0%), disgusted (0%), surprised (0%)}.

In our analysis, we consider the highest percentage value from the positive or negative emotion group for each input sentence of our datasets obtained from their system. Thus for those two sentences, the first one is considered as positive and the other one as negative according to the output given by *EmpathyBuddy*. Table III summarizes the accuracy obtained for Dataset A and Dataset B from the experimental runs of the system where the valence range to signal neutrality is ±3.5 for Dataset A. This resulting average performance gain of our system is 11.17% and 12.86% with regard to accuracy for these two datasets respectively, when compared to Liu et al. [8].

While our system outperforms Liu's system in this setting, we want to emphasize that Liu's system was not designed for sentiment recognition. Hence, a direct (fair) comparison was not possible.

TABLE III. ACCURACY COMPARISON METRICS BETWEEN EMPATHYBUDDY SYSTEM AND OURS

Dataset A		Dataset B	
Our System	Liu's System	Our System	Liu's System
82%	70.83%	91.53%	78.67%

IV. CONCLUSION

The system described in this paper proposes a novel method to recognize sentiment at the sentence level. The system first performs semantic processing and then applies rules to assign contextual valence to the linguistic components in order to obtain sentence- level sentiment valence. The system is well-founded because we have employed both cognitive and commonsense knowledge to assign prior valence to the words and the rules are developed following the heuristics to exploit linguistic features. We have conducted several studies using various types of data that demonstrate the accuracy of our system when compared to human performance as "gold standard". Moreover, it outperforms a state-of-the-art system (under simplifying assumptions). We also achieved better performance or almost similar performance while experimenting with machine learning approaches with the same datasets. In general terms the research aims at giving computer programs a skill known as emotional intelligence with the ability to understand human emotion and to respond to it appropriately. We believe that this linguistic approach to assessing sentiment from texts would strengthen human-computer interaction with fun. We also want to classify texts based on several emotion-types following the OCC emotion-model [12] and perform evaluations using online resources (e.g. blogs, news etc.).

REFERENCES

[1] Machinese Syntax, 2005, the official website, http://www.connexor.com/connexor/

[2] English Vocabulary, http://www.englishclub.com/vocabulary/

[3] B. Eriksson, "Sentiment classification of movie reviews using linguistic parsing". Accessed Oct. 2006. Available online. URL: http://pages.cs.wisc.edu/~apirak/cs/cs838/eriksson_final.pdf

[4] C. Fellbaum, (ed.), *WordNet: An Electronic Lexical Databases*, MIT Press, Cambridge: Massachusetts, 1999.

[5] S. Fitrianie and L. J.M. Rothkrantz, "Constructing Knowledge for Automated Text-Based Emotion Expressions", In *Proceedings of CompSysTech*, (June 15-16, Tarnovo, Bulgaria), 2006

[6] J. Kamps and M. Mar, "Words with Attitude", In *Proceedings of the First International WordNet conference*, (Jan. 21-25, Mysore, India), 2002.

[7] S. Knobloch, "Affective News- Effects of Discourse Structure in Narratives on Suspense, Curiosity, and Enjoyment While Reading News and Novels", *Communication Research* 31, 3, pp. 259-287, June 2004.

[8] H. Liu, H. Lieberman, and T. Selker, "A Model of Textual Affect Sensing using Real-World Knowledge", In *Proceedings of the Seventh International Conference on Intelligent User Interfaces*, (January 12-15, Miami, FL), ACM,2003, pp. 125-132.

[9] H. Liu and P. Singh, "ConceptNet: A Practical Commonsense Reasoning Toolkit", *BT Technology Journal* 22, 4, pp. 211-226, Oct. 2004, Kluwer Academic Publishers.

[10] R. Mihalcea and H. Liu, "A corpus-based approach to finding happiness, Computational approaches for analysis of weblogs", *AAAI Spring Symposium 2006*, available online at: http://web.media.mit.edu/~hugo/publications/papers/CAAW2006-happiness.pdf.

[11] T. Nasukawa and J. Yi, "Sentiment Analysis: Capturing Favorability Using Natural Language Processing", In *Proceedings of the 2nd international conference on Knowledge CAPture*, (Oct. 2-5, Sanibel Island, FL), 2003, ACM Press, pp. 70-77.

[12] A. Ortony, G.L. Clore, and A. Collins, *The Cognitive Structure of Emotions*, Cambridge University Press, 1988.

[13] B. Pang and L. Lee, "Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales". In *Proceedings of the 43rd Annual Meeting of the Association of Computational Linguistics, (ACL-05)*, Michigan, 2005, pp. 115-124.

[14] J.W. Pennebaker, M.R. Mehl, and K. Niederhoffer, "Psychological aspects of natural language use: Our words, our selves". *Annual Review of Psychology* 54, (2003), 547-577.

[15] J.W. Pennebaker, M.E. Francis, and R.J. Booth, "Linguistic inquiry and word count: LIWC (2nd ed.)", *[Computer software]*. Mahwah, NJ: Erlbaum.

[16] R. Picard, *Affective Computing*, MIT Press, Massachusetts, 1997.

[17] RSS feeds of Yahoo! News, http://news.yahoo.com/rss

[18] F. de Rosis and F. Grasso, "Affective natural language generation, In Paiva, A. (ed): *Affective Interactions, Towards a New Generation of Computer Interfaces"*, Lecture Notes in Computer Science (LNCS), 1814, (2000), 204-218, Springer-Verlag.

[19] J.A. Russell, J.A. Bachorowski, and J.M. Fernandez-Dols, "Facial and Vocal Expressions of Emotion", *Annual Review of Psychology* 54, (2003), 329-349.

[20] F. Sebastiani, "Machine Learning in Automated Text Categorization", *ACM Computing Surveys* 34, Vol. 1, pp. 1-47, March 2002.

[21] M.A.M Shaikh, H. Prendinger, and M. Ishizuka, "Assessing Sentiment of Text by Semantic Dependency and Contextual Valence Analysis", In *Proceedings of the International Conference on Affective Computing and Intelligent Interaction,*(ACII-07), Lisbon, Portugal, 2007. pp. 191-202.

[22] M. A. M Shaikh, M.T. Islam, and M. Ishizuka, "ASNA: An Intelligent Agent for Retrieving and Classifying News on the Basis of Emotion-Affinity", In *Proceedings of the International Conference on Intelligent Agents, Web Technologies and Internet Commerce*, (IAWTIC-06), Sydney, Australia, 2006. pp. 133-138.

[23] M.A.M Shaikh, H. Prendinger, and M. Ishizuka, "SenseNet: A Linguistic Tool to Visualize Numerical-Valance Based Sentiment of Textual Data", In *Proceedings of the International Conference on Natural Language Processing, (ICON-07)*, Hyderabad, India, January 2007, pages 147-152.

[24] O. Stock and C. Strapparava, "Getting Serious about the Development of Computational Humor". In *Proceedings of International Joint Conference of Artificial Intelligence*, (August 9-15, Acapulco, Mexico), Morgan Kaufmann, 2003, pp. 59-64.

[25] P. Subasic and A. Huettner, "Affect Analysis of Text Using Fuzzy Semantic Typing", *IEEE Transactions on Fuzzy Systems* 9, Vol. 4, pp. 483 – 496, Aug. 2001.

[26] Survey report, web-site, http://ita.co.jp/research/survey/

[27] A. Valitutti, C. Strapparava, and O. Stock, "Developing Affective Lexical Resources", *PsychNology Journal* 2, Vol. 1, pp. 61-83, 2004.

[28] J. Wiebe, R. Bruce, M. Bell, M. Martin, and T. Wilson, "A corpus study of evaluative and speculative language", In *Proceedings of the Second SIGdial Workshop on Discourse and Dialogue* - Volume 16, (September 1-2, Aalborg, Denmark), 2001, ACL, pp. 1-10.

[29] X. Zhe and A.C. Boucouvalas, "Text-to-emotion engine for real time internet communication". In *Proceedings of the third International Symposium on CSNDSP*, (July 15-17, Staffordshire University, UK), 2002, pp. 164-168.