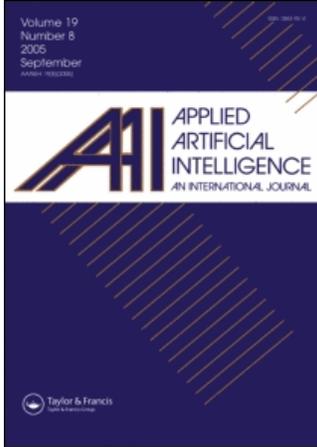


This article was downloaded by:[Shaikh, Mostafa Al Masum]  
On: 29 July 2008  
Access Details: [subscription number 795353780]  
Publisher: Taylor & Francis  
Informa Ltd Registered in England and Wales Registered Number: 1072954  
Registered office: Mortimer House, 37-41 Mortimer Street, London W1T 3JH, UK



## Applied Artificial Intelligence An International Journal

Publication details, including instructions for authors and subscription information:  
<http://www.informaworld.com/smpp/title~content=t713191765>

### SENTIMENT ASSESSMENT OF TEXT BY ANALYZING LINGUISTIC FEATURES AND CONTEXTUAL VALENCE ASSIGNMENT

Mostafa Al Masum Shaikh <sup>a</sup>; Helmut Prendinger <sup>b</sup>; Mitsuru Ishizuka <sup>a</sup>

<sup>a</sup> Department of Information and Communication Engineering, Graduate School of  
Information Science and Technology, University of Tokyo, Tokyo, Japan

<sup>b</sup> Digital Contents and Media Sciences Research Division, National Institute of  
Informatics, Tokyo, Japan

Online Publication Date: 01 July 2008

To cite this Article: Shaikh, Mostafa Al Masum, Prendinger, Helmut and Ishizuka, Mitsuru (2008) 'SENTIMENT ASSESSMENT OF TEXT BY ANALYZING LINGUISTIC FEATURES AND CONTEXTUAL VALENCE ASSIGNMENT', Applied Artificial Intelligence, 22:6, 558 — 601

To link to this article: DOI: 10.1080/08839510802226801  
URL: <http://dx.doi.org/10.1080/08839510802226801>

PLEASE SCROLL DOWN FOR ARTICLE

Full terms and conditions of use: <http://www.informaworld.com/terms-and-conditions-of-access.pdf>

This article maybe used for research, teaching and private study purposes. Any substantial or systematic reproduction, re-distribution, re-selling, loan or sub-licensing, systematic supply or distribution in any form to anyone is expressly forbidden.

The publisher does not give any warranty express or implied or make any representation that the contents will be complete or accurate or up to date. The accuracy of any instructions, formulae and drug doses should be independently verified with primary sources. The publisher shall not be liable for any loss, actions, claims, proceedings, demand or costs or damages whatsoever or howsoever caused arising directly or indirectly in connection with or arising out of the use of this material.

## SENTIMENT ASSESSMENT OF TEXT BY ANALYZING LINGUISTIC FEATURES AND CONTEXTUAL VALENCE ASSIGNMENT

**Mostafa Al Masum Shaikh<sup>1</sup>, Helmut Prendinger<sup>2</sup>, and Mitsuru Ishizuka<sup>1</sup>**

<sup>1</sup>*Department of Information and Communication Engineering, Graduate School of Information Science and Technology, University of Tokyo, Tokyo, Japan*

<sup>2</sup>*Digital Contents and Media Sciences Research Division, National Institute of Informatics, Tokyo, Japan*

□ *Text is not only an important medium to describe facts and events, but also to effectively communicate information about the writer's positive or negative sentiment underlying an opinion, or to express an affective or emotional state, such as happiness, fearfulness, surpriseneess, and so on. We consider sentiment assessment and emotion sensing from text as two different problems, whereby sentiment assessment is the task that we want to solve first. Thus, this article presents an approach to sentiment assessment, i.e., the recognition of negative or positive valence of a sentence. For the purpose of sentiment recognition from text, we perform semantic dependency analysis on the semantic verb frames of each sentence, and then apply a set of rules to each dependency relation to calculate the contextual valence of the whole sentence. By employing a domain-independent, rule-based approach our system is able to automatically identify sentence-level sentiment. A linguistic tool called "SenseNet" has been developed to recognize sentiments in text, and to visualize the detected sentiments. We conducted several experiments with a variety of datasets containing data from different domains. The obtained results indicate significant performance gains over existing state-of-the-art approaches.*

Recognizing or "sensing" affective information would benefit the development of text-based user interfaces since the words people use to express their feelings can be important clues to their mental, social, and physical state (Pennebaker, Mehl, and Niederhoffer 2003). Examples of such applications include the affective text analyzer (Hu and Liu 2004; Mihalcea and Liu 2006; Shaikh, Islam, and Ishizuka 2006b; Shaikh, Prendinger, and Ishizuka 2007a, 2007b; Knobloch, Patzig, Mende, and Hastall 2004; Sentiment! 2005; Pennebaker, Francis, and Booth 2001), the affective e-mail client (Liu, Lieberman, and Selker 2003), empathic chat (Zhe and Boucouvalas

2002), information and tutoring tools (Rosis and Grasso 2000), computational humor (Stock and Strapparava 2003), affective lexicon (Valitutti, Strapparava, and Stock 2004, Esuli and Sebastiani 2006), affective information recognizer (Mihalcea and Liu 2006; Kim and Hovy 2006; Koppel and Shtrimerberg 2004; Carenini, Ng, and Zwart 2005), and psycholinguistic analysis (Pennebaker et al. 2003; Kamps and Marx 2002). We expect that more are likely to appear with the increase of textual resources on the internet (e.g., blogs, reviews, etc.). We are interested in identifying positive and negative sentiment as well as emotions (e.g., happiness, sadness, etc.) conveyed through text. Our approach is based on the semantic relationship between textual components in a sentence and the computation of contextual valence of the used words. The scope of this article, however, is limited to sentiment assessment and sentiment visualization.

There are four main factors that distinguish our work from others. First, we integrated semantic processing of input text by performing dependency analysis of semantic verb frame(s) (SVF) of each sentence. The idea of a semantic verb frame is borrowed from FrameNet (Johnson et al. 2006). A frame in FrameNet (e.g., Apply\_heat) usually describes a common situation involving some roles defined as frame elements (FEs) (e.g., COOK, some FOOD, and a HEATING\_INSTRUMENT), and is evoked by words (e.g., bake, blanch, boil, broil, brown, simmer, steam, etc.). Frame-evoking words are called lexical units (LUs). In the simplest case, the frame-evoking LU is a verb and the FEs are its syntactic dependents. Similarly, the SVF in this case is composed of a verb word linked with its subject and object. The notion of SVF is to represent an event described in the input. Second, cognitive and common sense knowledge resources have been utilized to assign a prior valence to a set of words that leverage scoring for new words. Third, a set of rules to calculate contextual valence has been implemented to support word sense disambiguation. Finally, instead of using machine-learning (ML) or relying on text corpora, we followed a rule-based approach to assess the valence of each SVF reported in a sentence, and then assign an overall valence to the whole sentence(s) by applying dedicated rules. This paradigm of content analysis allows assessing sentiment (both *quantitative and qualitative scoring*) of texts of any genre (e.g., movie or product review, news articles, blogs, etc.) at the sentence level.

The remainder of the article is organized as follows. The next section provides a discussion of related works with an eye on the limitations of the existing approaches in textual affect sensing, and briefly motivates the approach from the perspective of cognitive psychology. The SenseNet architecture is the core of the article where we frame our approach by describing the architecture and implementation of the system. First we discuss our approach to semantic parsing of the input sentence(s). Then, we explain how different linguistic resources like WordNet (Fellbaum 1999),

and ConceptNet (Liu and Singh, 2004) are integrated to build the system's knowledge base. Next, we explain the formal underpinnings of the rules that compute the valence to indicate the positive, negative, or neutral sentiment of the input sentence(s). Finally, based on an example input sentence, we provide a detailed explanation of our algorithm by "walking through" the steps of the algorithm. The section on "System Evaluation" discusses the evaluation of our approach using standard datasets, and reports the results. The "Discussion" section describes strengths and weaknesses of our approach. Conclusions are drawn in the last section.

## BACKGROUND AND RELATED WORK

Sentiment has been studied at three different levels: word, sentence, and document level. There are methods to estimate positive and negative sentiment of words (Turney 2002; Esuli and Sebastiani 2005, 2006), phrases and sentences (Kim and Hovy 2006; Wilson, Wiebe, and Hoffmann 2005), and documents (Hu and Liu 2004). Previous approaches for assessing sentiment from text are based on one or a combination of the following techniques: keyword spotting (Zhe and Boucouvalas 2002), lexical affinity (Valitutti et al. 2004; Kim and Hovy 2005), statistical methods (Pennebaker et al. 2001, 2003), a dictionary of affective concepts and lexicon (Rosis and Grasso 2000), common sense knowledge base (Mihalcea and Liu 2006; Liu et al. 2003), fuzzy logic (Subasic and Huettner 2001), knowledge base from facial expression (Fitriani and Rothkrantz 2006), machine-learning (Gamon 2004; Kim and Hovy 2006; Wiebe, Wilson, and Cardie 2005; Koppel and Shtrimberg 2004; Sebastiani 2002), domain-specific classification (Nasukawa and Yi 2003; Koppel and Shtrimberg 2004), and valence assignment (Shaikh et al. 2007a, 2007b; Wilson et al. 2005; Polanyi and Zaenen 2004).

Some researchers proposed ML methods to identify words and phrases that signal subjectivity. For example, Wiebe and Mihalcea (2006) stated that subjectivity is a property that can be associated with word senses, and hence word sense disambiguation can directly benefit subjectivity annotations. Turney (2002) and Wiebe (2000) concentrated on learning adjectives and adjectival phrases, whereas Wiebe et al. (2005) focused on nouns. Riloff, Wiebe, and Wilson (2003) extracted patterns for subjective expressions as well. Machine-learning has been applied to various domains, such as movie reviews by Pang, Lee, and Vaithyanathan (2002) and Pang and Lee (2004), product reviews by Turney and Littman (2003) and Kim and Hovy (2006), and customer feedback reviews by Gamon (2004). In Pang et al. (2002), it is shown that a ML algorithm outperforms a simple term counting method. Much of the research up to now has focused on training machine-learning algorithms, such as support vector machines (SVMs),

to classify reviews. Pang et al. (2002) and Pang and Lee (2004) compared several ML algorithms and found that SVMs generally gave better results. Unigrams, bigrams, part of speech information, and the position of the terms in the text were used as features; however, using only unigrams was found to give the best results, with an accuracy of up to 83%. A variety of features was used with SVMs in an attempt to divide the data set not only into positive and negative, but also to give rankings of 1, 2, 3, and 4, where 1 means “not satisfied” and 4 means “very satisfied.” The proposed system performed fairly well at distinguishing classes 1 from 4, with about 78% accuracy. Separating classes 1, 2 from 3, 4 proved more difficult, with an accuracy of only 69%. These results were achieved when using the top 2000 features selected by log likelihood ratios. The research of Mullen and Collier (2004) introduced an approach called hybrid SVM, which brings together diverse sources of potentially pertinent information, including several favorability measures for phrases and adjectives and a knowledge of the topic of the text. Models using the features introduced are further combined with unigram models which have been shown to be effective in the past (Pang et al. 2002) and lemmatized versions of the unigram models. Experiments on movie review data from the Internet movie database demonstrated that hybrid SVMs which combine unigram-style, feature-based SVMs with those based on real-valued favorability measures, obtained superior performance. We observe that sentences typically convey affect through underlying meaning rather than affect words, and thus evaluating the affective clues is not sufficient in recognizing affective information from texts.

According to a linguistic survey (Pennebaker et al. 2003), only 4% of the words used in written texts carry affective content. This finding shows that using affective lexicons is not sufficient in recognizing affective information from text. It also indicates the difficulty of employing methods like machine-learning, keyword spotting, or lexical affinity (detailed criticisms are given in Liu et al. (2003) and Shaikh, Prendinger, and Ishizuka 2006a)). Statistical methods are suited for a psycholinguistic analysis (e.g., Pennebaker et al. 2003) of persons’ attitudes, social class, standards, etc. from documents rather than individual sentences. Fuzzy logic-based approaches assess input text by spotting regular verbs and adjectives that have preassigned affective categories (centrality and intensity), but ignore their semantic relationships. Similar to machine-learning, this technique cannot be used for analyzing smaller text units such as sentences. Senti-WordNet (Esuli and Sebastiani 2006) is a lexical resource that assigns to each synset of WordNet three sentiment scores, namely, positivity (i.e., P value), negativity (i.e., N value), and objectivity (i.e., O value), which indicate how positive, negative, and objective the terms contained in the synset are. For example, the adjective “estimable” has three senses. The output

given by SentiWordNet for the first sense is  $P = 0.75$ ,  $N = 0$ ,  $O = 0.25$ ; for the second sense is  $P = 0.625$ ,  $N = 0.25$ ,  $O = 0.125$ ; and for the third sense is  $P = 0$ ,  $N = 0$ ,  $O = 1$ . The values indicate that the first two senses express subjective opinion, and the third one involves objective evaluation. We can use the scoring provided by SentiWordNet if we can implement a technique to discern the sense of a word used in the context of the input text from all the senses returned by the given word. Although some works distinguish between different POSs of a word, the distinction between different senses of a word was never attempted. Therefore, we cannot apply the scoring provided by SentiWordNet for two reasons: firstly, due to the inability to decide the exact sense used in the input sentence, and secondly, due to having to consider all the words in the input subjectively.

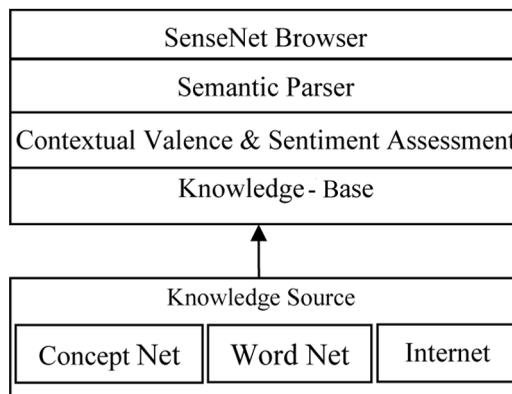
A number of researchers (e.g., Kamps and Marx [2002], Turney [2002], Wilson et al. [2005]) have explored the automatic learning of words and phrases with prior positive or negative valence. By contrast, we begin with a lexicon of words by calculating prior valence using WordNet (Fellbaum 1999) and ConceptNet (Liu and Singh 2004), and assign the contextual valence (Polanyi and Zaenen 2004) of phrases by applying a set of dedicated rules. Kim and Hovy (2006), Hu and Liu (2004), and Wilson et al. (2005) multiply or count the prior valence of opinion-bearing words of a sentence. They also consider local negation to reverse valence but they do not perform a deep analysis (e.g., semantic dependency), as our approach does. Nasukawa and Yi (2003) classify the contextual valence of sentiment expressions (as we do) and also expressions that are about specific items based on manually developed patterns and domain-specific corpora, whereas our approach is domain-independent. The use of domain-specific corpora for sentiment classification through feature extraction from evaluative text has shown very promising results regarding sentiment analysis of product reviews and blogs. For example, the method described by Carenini et al. (2005) applies feature extraction for products (i.e., digital camera and DVD) based on an existing unsupervised method. By including user-specified prior knowledge of the evaluated entity, the method turns the task of feature extraction into one of term similarity, thereby mapping crude features into a user-defined taxonomy of the entity's features. Thus, user-defined features provide a powerful and cost-effective means of complexity reduction, by truncating the long list of automatically extracted features into useful knowledge consisting of a nonredundant set of features. Such a method is useful in answering two key questions for product designers, planners, and manufacturers: "what product features are most frequently mentioned by customers?" and "do customers disagree on their evaluations of such features?". The answers can be found by evaluating the reports for how many times the feature is evaluated in the corpus, and by investigating how many times the evaluation is positive vs. negative.

Another system called OASYS (Cesarano et al. 2006) is an opinion analysis system that measures opinion coined as “public opinion” on a given topic extracting negative and positive phrases from the news articles. These types of systems require special tuning of data in order to build category-specific classifiers for each text domain (e.g., product review or movie review).

## SENSENET ARCHITECTURE

We concede that the analysis of favorable or unfavorable opinions, or emotion-affinity, is a task requiring emotional intelligence and a deep understanding of the textual context, involving common sense, domain knowledge, as well as linguistic knowledge. The interpretation of opinions is usually debatable, arguable, doubtful, subjective, and an idiosyncratic affair even for humans (Wiebe, Bruce, Bell, Martin, and Wilson 2001). Nevertheless, by proposing SenseNet, we will attempt a computational approach to solve this task. The compositional architecture of SenseNet is indicated in Figure 1. SenseNet maintains a knowledge base by employing three types of knowledge sources: WordNet 2.1 (Fellbaum 1999), ConceptNet 2.1 (Liu and Singh 2004), and the Internet. A set of rules has been implemented to compute contextual valence and to perform sentiment assessment. The semantic parser has been developed on top of a language parser (Machines Syntax 2007) and is utilized to perform semantic processing. The SenseNet browser shows the sentiment of each line of the input text by displaying numerical valence and icons. Subsequent sections explain the components in detail.

In a linguistic context, as in, e.g., in WordNet, the sense of a word is a given meaning of that word within a certain context. Similar to WordNet, the term “sense” in SenseNet refers to the contextual sense of each



**FIGURE 1** Architecture of SenseNet.

semantic verb frame(s) (Fellbaum 1999; Johnson et al. 2006) of a sentence, whereby each sense is represented by a lexical triplet consisting of a subject or agent, a verb, and an object. According to this naming convention, the input sentence “*We have submitted a paper to the conference and we are very optimistic*” involves two senses. They are based on the two verbs “submit” and “be,” and associated with two triplets [we, submit, paper] and [we, be, optimistic], respectively. The motivation for creating SenseNet is to utilize several linguistic resources (e.g., Language Parser, WordNet, ConceptNet, etc.) to construct “senses” based on the semantic verb frames of the input sentence(s) as the computational elements; assess the contextual valence of the sense(s); and finally, output a valence to indicate either positive or negative sentiment of the input sentence(s) by a graphical manner.

The system implements a pipelined design with the following phases: Parse, Process, Assess, and Visualize. Briefly, the parse phase implements semantic parsing, i.e., it performs dependency analysis on the words and outputs triplet(s) of subject, verb, and object according to each semantic verb frame of the input sentence(s). In the process phase, rules are applied to assign contextual valence to the triplet(s). In the assess phase an overall valence is assigned to each input sentence(s). Finally, the visualize phase, the SenseNet browser displays the sentiments of the input text using icons and symbols.

## Semantic Parser

For each input sentence, the semantic parser module outputs triplet(s) consisting of a subject or agent, a verb, and an object. Each member of the triplet may or may not have associated attribute(s) (e.g., adjective, adverb, etc.). Using the Machinese Syntax parser (2007), we first obtain

**TABLE 1** Triplet Output of Semantic Parsing for the Sentence Given Above

Senses processed by SenseNet	
<b>Triplet 1</b>	[[['Subject-Name:', 'raid', 'Subject-Type:', 'concept', 'Subject-Attrib:', ['A ABS: Israeli', 'N NOM SG: air']], ['Action-Name:', 'kill', 'Action Status:', 'Past Participle', 'Action-Attrib:', ['passive', 'time: Sunday', 'place: Lebanon']], ['Object-Name:', 'member', 'Object-Type:', 'person', 'Object-Attrib:', ['NUM: eight', 'A ABS: Canadian', 'N NOM: family', 'N NOM: vacationing']]]
<b>Triplet 2</b>	[[['Subject-Name:', 'raid', 'Subject-Type:', 'concept', 'Subject-Attrib:', [], ['Action-Name:', 'hit', 'Action-Status:', 'Past', 'Action-Attrib:', [], ['Object-Name:', 'town', 'Object-Type:', 'N NOM', 'Object-Attrib:', ['A ABS: Lebanese', 'place: border', 'N NOM: Israel']]]]
<b>Triplet 3</b>	[[['Subject-Name:', 'official', 'Subject-Type:', 'Object', 'Subject-Attrib:', ['A ABS: Canadian', 'A ABS: Lebanese']], ['Action-Name:', 'say', 'Action-Status:', 'Past', 'Action-Attrib:', [], ['Object-Name:', ' ', 'Object-Type:', ' ', 'Object-Attrib:', []]]]

XML-formatted syntactic and functional dependency information for each word of the input text, which constitutes the basis for generating the triplet(s). Since a new triplet is generated for each occurrence of a verb in the sentence, semantic parsing may extract more than one such triplet if multiple verbs are present in the sentence.

Basically, a triplet encodes information about “who is associated with what, where, and how” with a notion of semantic verb frame analysis. For example, the input sentence “*Eight members of a Canadian family vacationing in Lebanon were killed Sunday in an Israeli air raid that hit a Lebanese town on the border with Israel, Canadian and Lebanese officials said*”. produces three triplets as shown in Table 1.

### WordNet

WordNet (Fellbaum 1999) is a database of English words organized into synonym sets, whereby each word is linked by a small set of semantic relations, such as the synonym relation or the “is-a” hierarchical relation. The current version of WordNet (Version 2.1) contains 207,016 word-sense pairs and 78,695 polysemous senses. A sense, in the context of WordNet, is a distinct meaning that a word can assume. As a simple semantic network with words as the nodes, it can be readily applied to any textual input for query expansion, or determining semantic similarity. Thus, for an input word, we can obtain all the senses for a particular word which is usually not found in thesauri and dictionaries.

The SenseNet is employing WordNet 2.1 for two purposes. The primary purpose is to assign a numerical value (i.e., prior valence), denoting either positive or negative valence, to each of our enlisted words (i.e., “base list”) obtained from English Vocabulary (English Club 2006) based on manual investigation of senses of each word done by a group of judges (explained in the subsection Scoring a List of Verbs, Adjectives, and Adverbs). The secondary purpose relates to situations where a word is not found in the “base list.” Here, the system may automatically assign a valence for that word by first obtaining the synonyms of that word, and then screening the synonyms with respect to the “base list” for which numerical values are already assigned. Then the new word and its valence are inserted into the “base list.”

### ConceptNet

ConceptNet (Liu and Singh 2004) is a semantic network of common sense knowledge that currently contains about 1.6 million edges connecting more than 300,000 nodes. Nodes are semi-structured English fragments, interrelated by an ontology of 20 semantic relations encompassing

```

% conceptnet 2.0 mini-browser
rocket

BROWSE CONTEXT | PROJECTION | ANALOGY | GUESS CONCEPT | GUESS TOPIC | GUESS MOOD
==ConceptuallyRelatedTo==> space (3, 0)
==CapableOfReceivingAction==> launch from launch pad (2, 0)
==ConceptuallyRelatedTo==> gravity (2, 0)
==ConceptuallyRelatedTo==> more thrust (2, 0)
==ConceptuallyRelatedTo==> launch platform (2, 0)
==ThematicKLine==> space (2, 0)
==CapableOf==> do not travel with platform (1, 0)
==CapableOf==> contain fuel (1, 0)
==CapableOf==> fly up into sky (1, 0)
==PartOf==> engine (0, 1)
==CapableOfReceivingAction==> launch from launch platform (1, 0)
==CapableOf==> roar from its pad into space (1, 0)
==UsedFor==> launch space shuttle into orbit (1, 0)
==CapableOf==> roar (0, 1)

```

FIGURE 2 ConceptNet output for the concept “rocket”.

the spatial, physical, social, temporal, and psychological aspects of everyday life. ConceptNet is generated automatically from the 700,000 sentences of the Open Mind Common Sense (OMCS) Project, which were gathered from worldwide web-based collaboration with over 14,000 authors. A robust approach for weighting knowledge is implemented, which scores each binary assertion based on how many times it occurred in the OMCS corpus, and on how well it can be inferred indirectly from other facts in ConceptNet. One can consider ConceptNet as an extension of a model of purely lexical items with atomic meaning to higher-order compound concepts, which compose an action verb with one or two direct or indirect arguments. It also extends WordNet’s list of semantic relations to a repertoire of 20 semantic relations including, for example, EffectOf (causality), SubeventOf (event hierarchy), CapableOf (agent’s ability), PropertyOf, LocationOf, and MotivationOf (affect). Moreover, the knowledge in ConceptNet is of a more informal, defeasible, and practically valued nature.

In the SenseNet, we have employed ConceptNet to retrieve all applicable semantic relationships of the input concept with other concepts. This is necessary to assign prior valence of a concept. (In the subsection on Scoring of Nouns, we will explain how we process the output of ConceptNet.) By way of example, Figure 2 shows the semantic relationships obtained for the concept “rocket” with other concepts.

### The Knowledge Base

A common approach to sentiment assessment is to start with a set of lexicons whose entries are assigned a prior valence indicating whether a word, independent of context, evokes something positive or something

negative (Wilson et al. 2005). For instance, “destroy” usually bears a negative connotation, whereas “develop” typically has a positive connotation. Cognitive and common sense knowledge resources have been utilized to assign prior valence to the lexicon entries, and the resources also leverage scoring of new words, as will be explained in subsequent subsections.

The system maintains several lists of words having such prior valence. The “base list” is a list of verbs, adjectives, adverbs together with their prior valence, which is assigned based on WordNet. The “concept list” is a list of nouns, whose prior valence is calculated using ConceptNet. The “entity list” contains the named entities (e.g., kofi annan, ipod, etc.) for which ConceptNet fails to assign a prior valence. The prior valence of such named entities is assigned using online resources. Since we are incorporating different resources to assign prior valence to the words, the question of “reliability” of the assigned score might arise. We will briefly discuss this issue in the discussion section.

#### *Scoring a List of Verbs, Adjectives, and Adverbs*

A group of eight judges has manually counted the number of positive and negative senses of each word of the initial “base list” of verbs, adjectives, and adverbs according to the contextual explanations of each sense found in WordNet 2.1. A judge’s score of a verb is stored as the following format:

verb-word : <positive-sense count, negative-sense count, prior valence,  
prospective value, praiseworthy value >

The prior valence, prospective and praiseworthy values, indicate the lexical affinity of the verb with respect to “good” or “bad,” “desirable” or “undesirable,” and “praiseworthiness” or “blameworthiness,” respectively. Prospective and praiseworthy values of the verb words are not used in the system described in this article. We use those values in another system, where we aim to recognize more fine-grained emotions like “happy,” “sad,” “relief,” etc.

We will explain the scoring procedure by an example. For the word “kill,” WordNet 2.1 outputs 15 senses as a verb and each of the senses is accompanied by at least an example sentence or explanation to clarify the contextual meaning of the verb. Each judge reads each meaning of the sense and decides whether it evokes positive or negative sentiment. For example, for the word “kill,” one judge has considered 13 senses as negative and 2 senses as positive, which are stored in the scoring sheet. In this manner, we initially collected the scores for 723 verbs, 205 phrasal verbs, 237 adjectives related to shape, time, sound, taste/touch, condition,

**TABLE 2** Sample List of Verbs with Associated Prior Valence

Verb word	Prior valence
Amuse	3.750
Attack	-3.333
Battle	-5.000
Kill	-3.167
Thank	5.000
Wish	4.643
Yell	-1.250

appearance and 711 adjectives related to emotional affinity and 144 adverbs.

Equation (1) assigns a prior valence (i.e., a value between  $-5$  and  $5$ ) to each selected word:

$$pv(w) = \frac{\sum_{i=1}^m \left( \left( \frac{p_i - n_i}{N} \right) * 5.0 \right)}{m}. \quad (1)$$

Here,  $pv(w)$  = prior valence of word  $w$ , whereby  $-5 \leq pv(w) \leq +5$

$m$  = number of judges (in this case,  $m = 8$ ).

$p_i$  = the number of positive senses assigned by  $i$ th judge, for word  $w$ .

$n_i$  = the number of negative senses assigned by  $i$ th judge, for word  $w$ .

$N_i$  = total number of senses counted by  $i$ th judge for word  $w$ .

A subset of verbs (e.g., like, love, hate, kiss, etc.) of the “base list” is marked by a tag named  $\langle \text{affect} \rangle$  to indicate that these verbs have strong affective connotation regarding preference or dislike. This tagging is done manually according to the semantic labels (i.e., a-labels) of WordNet-Affect (Valitutti et al. 2004). To measure interagreement among judges, we used Fleiss’ Kappa statistic (Fleiss 1971). The Kappa value for the prior valence assignment task for the “base list” is reliable ( $\kappa = 0.914$ ). Moreover, our scoring resembles the EVA function (Kamps and Marx 2002) score that assigns values to a word based on the minimal path lengths from adjectives “good” and “bad.” A word not present in the annotated list is scored by calculating the average valence of its already scored synonyms obtained from WordNet. An excerpt from the verb database is given in Table 2.

### Scoring of Nouns

Since manual scoring is a tedious job and the number of nouns is usually greater than the count of the words in “base list,” we employed ConceptNet to assign prior valence to nouns in an automatic manner. A value

from  $[-5, +5]$  is assigned as the prior valence to an input noun or concept (we use “noun” and “concept” synonymously). If a concept is not present in the “concept list,” the system performs the following operations to assign prior valence to a concept. First, the system retrieves all other concepts that are semantically connected to the input concept using ConceptNet. For example, to assign valence to a concept  $C$ , the system collects all concepts  $Con_1, Con_2, \dots, Con_n$ , which are, respectively, connected to  $C$  with a specific semantic relationship like  $R_1, R_2, \dots, R_m$ . ConceptNet defines 20 such relationship types between concepts.

For the processing, the returned concepts are separated into two lists depending on the type of semantic relationships. The entries in the first list correspond to relationships like “IsA,” “DefinedAs,” “MadeOf,” “PartOf,” etc. and the entries in the second list correspond to relations like, “CapableOf,” “UsedFor,” “CapableOfReceivingAction,” “SubEventOf,” etc. Of the two groups, the first one indicates associated concepts that are basically nouns, and the second one indicates the actions (i.e., verb words) that the input concept can either perform or receive. The first list is matched against the “concept list,” and a maximum number of five concepts are considered for faster processing. The average of the prior valence values of the found concepts is assigned as the prior valence of the “to be scored,” concept. If this procedure cannot assign a nonzero value, a similar procedure is performed considering the second list and the scored verbs of the “base list.” The system considers the input concept as a named entity if the second procedure fails to assign a nonzero value as the prior valence of the “to be scored” concept. If a nonzero valence is obtained, the input concept and its prior valence are inserted into the “concept list” for future use.

Let us look at an example. Initially, for the concept “doctor,” the system failed to find a prior valence in the existing scored list of nouns. Here, the following two lists are obtained by applying the explained procedures and ConceptNet:

```
related.concept.list = ['person', 'smart person', 'human', 'conscious being',
'man', 'wiley bandicoot', 'clever person', 'dentist', 'pediatrician', 'surgeon',
'physician', 'veterinarian', 'messy handwriting', 'study medicine', 'job']
related.action.list = ['examine', 'help', 'look', 'examine patient',
'help sick person', 'wear', 'prescribe medicine', 'treat', 'prescribe',
'wear white coat', 'look at chart', 'save life', 'heal person', 'take care']
(the list is truncated due to space limitations)
```

In this case the system first processed the “related.concept.list”, and failed to assign a nonzero value because initially the “concept list” did

not have the score for those concepts in “related.concept.list.” Therefore, the second list, “related.action.list,” is processed and from the second list the system returned the value 4.21 by averaging the scores of the verbs, “examine (4.50)”; “help (5.00)”; “wear (2.57)”; “prescribe (4.27)”, and “treat (4.69).” Hence, the value 4.21 is assigned as the prior valence for the concept “doctor” and stored for future use. Instead of performing manual scoring of verbs, adjectives, and adverbs, we initially scored about 4500 concepts using the procedure explained above. The “concept list” is maintained to speed up the processing time since the system would otherwise have to invoke ConceptNet and perform scoring every time for a concept (i.e., noun word).

### *Scored List of Named Entity*

The system maintains a list named “entity list” that contains prior valence of named entities. We did not use any named entity recognizer to identify a named entity, and hence make the simplifying assumption that anything for which ConceptNet fails to assign a nonzero value is a named entity. The information of an entity is stored in the following format:

named entity:<concept, concept valence, general-sentiment, prior valence>

The attribute “concept” indicates a noun that describes the named entity in terms of “is a kind of” or “conceptually related to” type relationships. Attribute “concept valence” indicates the prior valence of the concept. For example, for the sentence, “*President George Bush spoke about the ‘Global War on Terror.’*”, the system signals “George Bush” as a named entity because it failed to assign a nonzero valence using ConceptNet. However, based on the output of the semantic parser, the system finds the noun “president” as an attribute associated with this named entity. Hence, for this named entity the system considers “president” as the “concept” attribute and from the ConceptNet system gets the prior valence for “president” as +2.75. If the system fails to receive any such noun attribute associated with the named entity, the system assumes an abstract concept named “person” assuming to have a “conceptually related to” type relationship. In this manner, it is attempted to extend the scope of ConceptNet by incorporating real-world knowledge.

The attribute “general-sentiment” contains either a negative (−1) or a positive (+1) value based on the value of the prior valence towards the named entity. To assign “general-sentiment” as well as prior valence we have developed a tool that can extract sentiment from Opinmind<sup>1</sup> (2006). Opinmind is a web-search engine that has a sentiment scale named “Sentimeter” which displays the relative number of positive and negative opinions expressed by people on anything regarding one’s views on politics

and current events. It also finds what people think about products, brands, and services by mining the opinion-bearing texts of people's blogs. Opinmind exercises no editorial judgment when computing "Sentimeter" values. For example, ConceptNet fails to assign a valence to "George Bush" or "Tokyo University." From Opinmind we obtain 37% positive, and 63% negative opinion regarding the named entity "George Bush." Similarly, for the input "Tokyo University" we obtain 100% positive, 0% negative opinion. From the obtained values we set the "general-sentiment" and "prior valence" as  $-1$  and  $-3.15$  (considering the maximum of the absolute value of the votes in the scale of 5) for "George Bush" and similarly for "Tokyo University" the values are  $+1$  and  $+5.1$ . Hence these are stored as: George Bush: <President,  $+2.752$ ,  $-1$ ,  $-3.15$ >; Tokyo University: <school,  $4.583$ ,  $+1$ ,  $+5.0$ >. Initially, a list of 2300 entries is manually created and scored using Opinmind. This list grows automatically whenever the system detects a new named entity. Since the service provided by Opinmind is presently suspended, as an alternative the technique mentioned in Grefenstette, Qu, Evans, and Shanahan (2004) is adopted to assign prior valence to a named entity exploiting web search result. Based on the method of Grefenstette et al. (2004) four affect classes of both negative and positive semantic axes are considered. The classes of positive axis are, "praise," "pleasure," "pride," and "comfort." Alternatively, the classes of negative axis are "slander," "pain," "humility," and "irritation." Each of the classes is represented by a bag of words as mentioned in Grefenstette et al. (2004). This approach can be considered as an extension of Turney and Littman's (2003) technique to other semantic classes relying on finding co-occurrences with the bag of words of the classes to measure the distance among the semantic classes. These distances are considered to automate the assignment of affect class centrality (i.e., either negative type or positive type in this case) of a named entity. In this manner, the prior valence of a named entity is calculated. Usually the value of "general sentiment" is idiosyncratic and arguable. If the valence sign of the "concept valence" and "general sentiment," (e.g., President [ $+2.752$ ], George Bush [ $-1$ ]) differs from each other, the system considers this as an ambiguity and assigns

**TABLE 3** Sample List of Scored Named Entities

Named entity	Concept	Concept valence	General sentiment	Prior valence
Bin Laden	War	$-4.625$	$-1$	$-3.10$
George Bush	President	$2.752$	$-1$	$-3.15$
Discovery	Shuttle	$3.984$	$+1$	$+4.25$
Kofi Annan	Person	$2.562$	$-1$	$-4.50$
Microsoft	Software	$4.583$	$-1$	$-2.65$
NASA	Space	$3.784$	$+1$	$+3.80$

neutral valence to the sentence referring that named entity. An excerpt from the named entity database is given in Table 3 to illustrate the idea.

### Contextual Valence Assessment

Before explaining the contextual valence assignment algorithm, we first discuss its underlying data structure.

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

**Processing Elements.** We assume the input is a paragraph  $P$ , containing  $n$  sentences, such that  $P = \langle S_1, S_2, \dots, S_i, \dots, S_n \rangle$  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  $T_j$ , i.e.,  $S_i = \langle T_1, T_2, \dots, T_j, \dots, T_m \rangle$ , whereby  $1 \leq j \leq m$ . A triplet  $T_j$  has the following form:  $\langle \text{actor, action, concept} \rangle$ . The triplet elements “actor” and “concept” have the following form:  $\langle \text{name, type, attribute} \rangle$ . The action has the form  $\langle \text{name, status, attribute} \rangle$ . An attribute is either an empty set or nonempty 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:

$$\begin{aligned}
 T_1 &= \langle \langle \text{President, Concept, \{the\}}, \langle \text{call, past, \{time: Tuesday,} \\
 &\quad \text{dependency: to} \rangle \rangle, \langle \text{discovery, Named Entity, \{the, space, shuttle\}} \rangle \rangle \\
 T_2 &= \langle \langle \text{President, Concept, \{the\}}, \langle \text{wish, infinitive, \{dependency: and\}}, \\
 &\quad \langle \text{astronaut, Concept, \{the, adv: well\}} \rangle \rangle \rangle \\
 T_3 &= \langle \langle \text{President, Concept, \{the\}}, \langle \text{congratulate, infinitive, \{dependency:} \\
 &\quad \text{and\}} \rangle \rangle, \langle \text{astronaut, Concept, \{goal: space walk\}} \rangle \rangle \\
 T_4 &= \langle \langle \text{President, Concept, \{the\}}, \langle \text{invite, infinitive, \langle \text{astronaut, Concept,} \\
 &\quad \langle \text{place : white house} \rangle \rangle \rangle \rangle
 \end{aligned}$$

**Knowledge Base.** The knowledge base of the system has been previously discussed. 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, the affective verb (AV) group and the nonaffective verb (V) group. The verbs having the tag  $\langle \text{affect} \rangle$  in the knowledge base are members of AV. Both AV and V are further partitioned into positive ( $AV_{\text{pos}}, V_{\text{pos}}$ ) and negative ( $AV_{\text{neg}}, V_{\text{neg}}$ ) groups on the basis of their prior valence. Similarly, adjectives (ADJ), adverbs (ADV), concepts (CON) also have positive and negative groups indicated

by  $ADJ_{pos}$ ,  $ADJ_{neg}$ ,  $ADV_{pos}$ ,  $ADV_{neg}$ ,  $CON_{pos}$ , and  $CON_{neg}$ , 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  $NE_{ambi}$ .

**Algorithm.** The core algorithm underlying our system can be summarized as follows. Input,  $P$ , is a paragraph that is a sequence of sentences. Output of the system is  $V$  that indicates valence values for each corresponding sentence. For each sentence, the following steps are performed. The pseudo-code of the algorithm for contextual valence assignment (i.e., function *getValence()*) is given in Appendix A.

First, the triplet representation (i.e., a set of triplets) of the sentence is obtained from the semantic parser. A triplet is basically consisting of a subject, verb, and object where each of them might have associated attributes like adverb, adjective, or nominative noun. To indicate the dependency relationship between two adjacent triplets the parser outputs 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 the input sentence. At present, for simplicity the dependency relationships are grouped into two types, namely, “to-dependency” (i.e., “dependency: to”) and “not\_to-dependency” (i.e., all others except “dependency: to”).

Second, all the triplets obtained from the input sentence are processed to assign a valence value to the sentence. This procedure involves the following steps: (1) Rules are applied to assign contextual valence to the subject, verb, and object of the triplet considering their attributes (i.e., adverb, adjective); (2) Conditionality, negation, and previously assigned contextual valence values are considered to assign a contextual valence to the triplets. Thus each triplet is assigned a contextual valence; (3) The dependency relationships (if any) among the adjacent triplets are considered and resultant valence values are assigned according to the “dependency processing” algorithm mentioned in the subsection on sentiment assessment. For the two types of dependencies, different sets of rules are applied to calculate resultant valence for two interdependent triplets. Finally, a valence is calculated for the input sentence from those resultant valence values. In this procedure, valence values are assigned to all the sentences of the input paragraph.

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

- $ADJ_{pos} + (CON_{neg} \text{ or } NE_{neg}) \rightarrow \text{neg. Valence}$  (e.g., strong cyclone; nuclear weapon)
- $ADJ_{pos} + (CON_{pos} \text{ or } NE_{pos}) \rightarrow \text{pos. Valence}$  (e.g., brand new car; final exam)

- $ADJ_{neg} + (CON_{pos} \text{ or } NE_{pos}) \rightarrow \text{neg. Valence}$  (e.g., broken computer; terrorist group)
- $ADJ_{neg} + (CON_{neg} \text{ or } NE_{neg}) \rightarrow \text{neg. Valence}$  (e.g., ugly witch; scary night)

Note that the sign of the valence switches because of 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:

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

Hence, the rules to resolve the ambiguity are:

- $ADV_{neg}\text{-except} + (AV_{neg} \text{ or } V_{neg}) \rightarrow \text{pos. Valence}$  (e.g., rarely forget; hardly hate)
- $ADV_{neg}\text{-not except} + (AV_{neg} \text{ or } V_{neg}) \rightarrow \text{neg. Valence}$  (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:

- $\text{Neg. Action Valence} + \text{Pos. Object Valence} \rightarrow \text{Neg. Action-Object Pair Valence}$  (e.g., kill innocent people, miss morning lecture, fail the final examination, etc.)
- $\text{Neg. Action Valence} + \text{Pos. Object Valence} \rightarrow \text{Pos. Action-Object Pair Valence}$  (e.g., quit smoking, hang a clock on the wall, hate the corruption, etc.)
- $\text{Pos. Action Valence} + \text{Pos. Object Valence} \rightarrow \text{Pos. Action-Object Pair Valence}$  (e.g., buy a brand new car, listen to the teacher, look after you family, etc.)

- Pos. Action Valence + Neg. Object Valence  $\rightarrow$  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 ( $AV_{pos}$ ,  $AV_{neg}$ ) which use the following rules to assign contextual valence for an affective verb:

- $AV_{pos} + (\text{pos. or neg. Object Valence}) = \text{pos. Action-Object Pair Valence}$  (e.g., I like romantic movies. She likes horror movies.)
- $AV_{neg} + (\text{neg. or pos. Object Valence}) = \text{neg. Action-Object Pair Valence}$  (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:

- $(CON_{pos} \text{ or } NE_{pos}) + \text{Pos. Action-Object Pair Valence} \rightarrow \text{Pos. Triplet Valence}$  (e.g., the professor explained the idea to his students.)
- $(CON_{pos} \text{ or } NE_{pos}) + \text{Neg. Action-Object Pair Valence} \rightarrow \text{Neg. Triplet Valence}$  (e.g., John rarely attends the morning lectures.)
- $(CON_{neg} \text{ or } NE_{neg}) + \text{Pos. Action-Object Pair Valence} \rightarrow \text{Tagged Negative Triplet Valence}$  (e.g., the robber appeared in the broad day light.) to process further.
- $(CON_{neg} \text{ or } NE_{neg}) + \text{Neg. Action-Object Pair Valence} \rightarrow \text{Neg. Triplet Valence}$  (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 triplets with a “dependency: and” attribute in the first triplet indicating that the first triplet has an “*and relationship*” with the second one. Of the two triplets, the first one is assigned to “tagged negative triplet valence” for the negative valence actor “*robber*” with a positive “action-object pair valence” for “*arrive, car.*” The other triplet is assigned with a “negative triplet valence” for having actor (“*robber*”) and “action-object pair valence” for “*mug, store-keeper*” as negative. So in this case, we notice that a negative valence actor is associated with a positive and negative “action-object pair.” For such cases, our simplified heuristic is that if a negative valenced actor is associated with at least one “negative

action-object pair,” the tagged output is considered as negative and the resultant valence is made negative. But if a negative valenced actor is associated with all positively scored “action-object pair” the “tagged negative triplet valence” is set to positive and the resultant valence is made positive. For example, “*The kidnapper freed the hostages and returned 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.” 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 Hu and Liu (2004) and Wilson et al. (2005).

### Sentiment Assessment

In the previous subsection we described how valence is assigned to triplets. Now we explain how sentiment (i.e., assessing contextual valence of the triplets) is assessed for a sentence. It is previously mentioned that from the semantic parser, two types of dependencies are tagged to indicate the dependency between two triplets. The system invokes a function (*process-TripletLevelContextualValence()*) to process the dependencies among the triplets and set the contextual valence of those triplets. The algorithm of this function is described below:

For the two triplets,  $T_1$  and  $T_2$  where  $T_1$  has a “to\_dependency” relationship with  $T_2$ , the contextual valence of the triplets are calculated according to the following rules:

- Contextual Valence Value =  $(\text{abs}(\text{valence of } T_1) + \text{abs}(\text{valence of } T_2))/2$
- Pos. valence of  $T_1$  + Pos. valence of  $T_2 \rightarrow$  Pos. Contextual Valence (e.g., I am interested to go for a movie.)
- Neg. valence of  $T_1$  + Pos. valence of  $T_2 \rightarrow$  Neg. Contextual Valence (e.g., It was really hard to swim across this lake.)
- Pos. valence of  $T_1$  + Neg. valence of  $T_2 \rightarrow$  Neg. Contextual Valence (e.g., It is easy to catch a cold at this weather.)
- Neg. valence of  $T_1$  + Neg. valence of  $T_2 \rightarrow$  Pos. Contextual Valence (e.g., It is difficult to take bad photo with this camera.)

Similarly, the rules to deal with “not\_to\_dependency” relationship are:

- Contextual Valence Value =  $(\text{abs}(\text{valence of } T_1) + (\text{valence of } T_2))/2$
- Pos. valence of  $T_1$  + Pos. valence of  $T_2 \rightarrow$  Pos. Contextual Valence (e.g., they got married and lived happily.)

- Neg. valence of  $T_1$  + Pos. valence of  $T_2 \rightarrow$  Pos. Contextual Valence (e.g., John was not a regular student but he finally scored good grades.)
- Pos. valence of  $T_1$  + Neg. valence of  $T_2 \rightarrow$  Neg. Contextual Valence (e.g., the movie was very interesting but at the end it became monotonous.)
- Neg. valence of  $T_1$  + Neg. valence of  $T_2 \rightarrow$  Neg. Contextual Valence (e.g., I feel very sad when my paper gets rejected.)

The pseudo-code of the function *processTripletLevelContextualValence()* is given in Appendix A. This function returns a list, namely “contextual-Valence” which contains valence values of the triplets after processing their dependencies. The average of the absolute values of the list “contextual-Valence” is assigned as the “sentimentScore” for the sentence, S. The “valenceSign” is set +1 if the count of positive values in the list is greater than the number of negative ones and vice versa. If both negative and positive counts are equal then +1 is set if the sign of the maximum value considering the absolute values of the list is positive, otherwise -1 is set. The value of “sentimentScore” is multiplied with “valenceSign” to get “sentenceValence” and this is the valence the system finally for the input sentence. According to the scoring system the range of “sentenceValence” is  $\pm 15$  since the maximum and minimum valence of a triplet can be 15 and -15, respectively.

The above idea is further explained by an example of how contextual valence values are assigned to the triplets of the input sentence, “*Tropical storm Bilis killed at least 48 people and injured hundreds as it churned across China’s south-east, toppling houses and forcing authorities to evacuate a prison and thousands of villagers.*”

SenseNet detected the following seven triplets for the input sentence

Triplet 1: [‘Bilis {tropical, storm}’, ‘kill {dependency: and}’, ‘people {at least, 48}’],

Triplet 2: [‘Bilis’, ‘injure {dependency: as}’, ‘people, {hundreds}’],

Triplet 3: [‘Bilis’, ‘churn across {dependency: and}’, ‘china {south-east}’],

Triplet 4: [‘Bilis’, ‘topple {dependency: and}’, ‘house’],

Triplet 5: [‘Bilis’, ‘force {dependency: to}’, ‘authority’],

Triplet 6: [‘authority’, ‘evacuate {dependency: and}’, ‘prison’],

Triplet 7: [‘authority’, ‘evacuate’, ‘villagers’]

All the attributes of the triplets are not shown due to space limitations. In the first triplet, the subject “Bilis” is a named entity which will be evaluated as the concept “storm” because it appears as a noun attribute of the subject (i.e., “Bilis”). In the subsequent triplets, the pronoun “it” as the subject has been replaced by the previously found subject “Bilis.” Due to

the presence of a noun (i.e., “authority”) as the object in the fifth triplet and the presence of a verb (i.e., “evacuate”) with a “dependency: to” relationship without having a direct subject, semantic parser considers “authority” as the subject for the sixth and seventh triplet. The sixth and seventh triplet have the same verb connecting two objects with an “and” relationship.

From the knowledge base we get the following prior valence for the words found in the example sentence:

“storm” : -3.394; “tropical” : 2.861; “kill” : -3.937; “people” : 2.5;  
 “injure” : -3.634; “churn across” : -3.696; “china” : 3.450;  
 “south-east” : 0.0; “topple” : -3.324; “house” : 5.0; “force” : 2.985;  
 “authority” : 3.196; “evacuate” : -2.694; “prison” : 0.588; “villager” : 3.812.

According to the algorithm (*getValence()* in Appendix A), the system prepares the list of triplets along with the dependency relationships (i.e., “tripletResult”) as following: {(-10.650, true, “dependency: and”), (-10.343, true, “dependency: as”), (-11.359, true, “dependency: and”), (-12.537, true, “dependency: and”), (-10.394, true, “dependency: to”), (-6.478, true, “dependency: and”), (-9.702, false, null)}. The numerical values shown in the list indicates the valence of the corresponding triplets (i.e., “triplet-Valence”). The dependencies among the triplets and the valence of the triplets (i.e., “tripletResult”) are processed (by the function *processTriplet-LevelContextualValence()*) to set the contextual valence of those triplets. According to the aforementioned algorithm of this function, the following list (i.e., “ContextualValence”) of values is obtained: [-10.496, -10.851, -11.948, -11.465, +9.242]. The fifth value of the list is positive because of the rule of having two negative triplets connected with “to.dependency” relationship. On processing this list of values the “valenceSign” is set negative because most of the values are negative and the “sentimentScore” is obtained as 10.80 for this sentence. Finally the “sentenceValence” is outputted as -10.80 indicating that the sentence bears a negative sentiment. Similarly for the sentence “*It is difficult to take bad photo with this camera,*” the “sentenceValence” is obtained as +12.251 indicating the sentence expressing a positive sentiment.

### SenseNet Browser

The SenseNet browser graphically visualizes each sentence in terms of the triplets and their associated valence values. SenseNet Browser is the front-end user interface for SenseNet and it is written in C#. It takes the input from the users and sends it to the back-end python implemented

program for analysis through TCP/IP socket connection. As shown in Figure 3, the browser has two panels for user interaction, namely, “input panel” and “sentiment browse panel.” In “input panel” a chunk of text can be input and clicking on the “analyze” button sends the text to the back-end python application to process it and finally receives the output. By clicking on the “visualize” button an ordered iconic representation of underlying sentiment of each input sentence(s) is displayed on the “sentiment browse panel” corresponding to the order of appearance of the sentences. The browser also has two other panels, namely “valence analysis panel” and “legend panel”. A click on any of the icons of “sentiment browse panel” is considered as the user’s request to show the analysis of the sentiment for that particular sentence represented by that icon. “Valence analysis” panel then shows the triplets and the valences associated with those. The “legend panel” explains the different icons and symbols used by the browser. SenseNet classifies sentences into three classes, namely, negative, positive, and neutral. According to the performed experiment (see System Evaluation) it is decided that for a sentence whose valence is between the ranges of  $\pm 3.5$  it is decided as a neutral sentence.

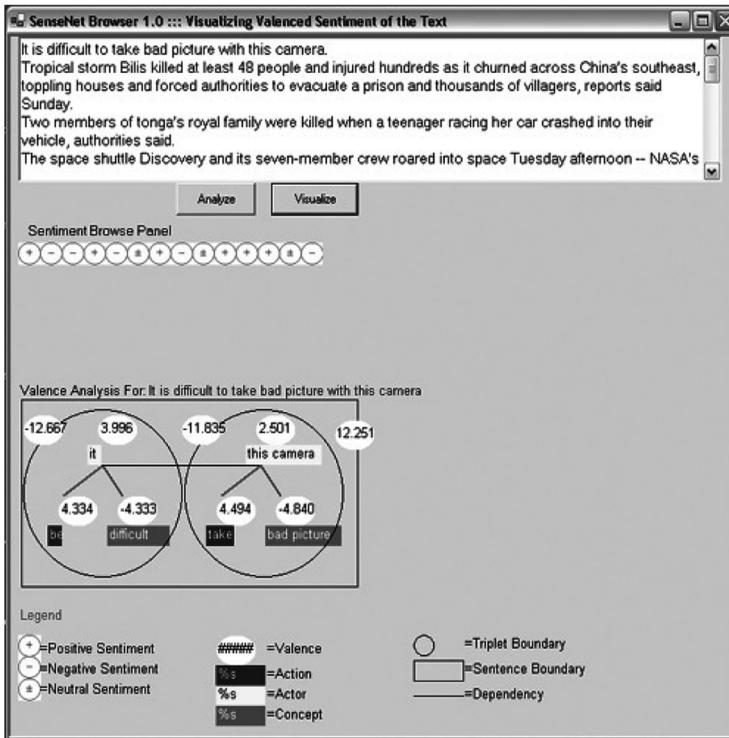


FIGURE 3 Interface and sample output of SenseNet browser.

**TABLE 4** Symbols Used in SenseNet Browser

Symbol	Explanation
	Indicates positive sentiment of a sentence.
	Indicates negative sentiment of a sentence.
	Indicates neutral sentiment of a sentence.
	White circle with a numerical value inside indicates the contextual valence of a linguistics components (e.g., noun, verb, sense, sentence)
	Text in black color filled rectangle indicates an action of a triplet.
	Text in white color filled rectangle indicates an actor of a triplet.
	Text in gray color filled rectangle indicates an object of a triplet.
	Black bordered circle represents a triplet.
	Black bordered rectangle represents the boundary of a sentence containing all the triplets.
	Black line connecting two actors indicates the interdependency between two triplets.
	Black lines connection action and concept/object with actor indicates the connectivity within a triplet.

Figure 3 shows an example of output obtained by SenseNet. The interface indicates that it has processed 14 sentences of which there are six positive, five negative, and three neutral sentiment carrying sentences. This is represented by the line of the circles with embedded polarity signs. Clicking on the first circle, the valence analysis for that sentence is shown. This type of browser will be helpful to readily identify and visualize the positive, negative, or neutral sentiment-bearing sentences from the textual data like product reviews or users' comments, blogs posts, e-mail contents, etc. Moreover, the iconic representation of underlying sentiment of the input text will help a user to grasp the sentimental perception (i.e., negative, positive, or neutral) of the input text in an easy manner. The idea of this browser might be extended to a multi-document level (e.g., a set of e-mails etc.) where the iconic representation of the "sentiment browse panel" would be produced based on the overall sentiments of the input documents. Such information visualization will help to filter contents quickly and easily.

The SenseNet browser uses several symbols to represent the visualization and analysis of the sentiment of texts. Table 4 explains the symbols.

## SYSTEM EVALUATION

We intend to evaluate our system both at the sentence level and paragraph (or document) level. To this end, we performed a 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).

## The Datasets

We use four datasets to test our method of sentiment assessment for both sentence and paragraph (or document) 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! 2007), and e-mail correspondences. It was scored by 20 human judges according to positive, negative, and neutral sentiment affinity by an online survey.<sup>2</sup> The judges were instructed to log in 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 the 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 5 judges, and as positive by 1 judge. Since the majority of the judges voted this sentence as a negative sentence, the sentence is considered a negative sentence in our “gold standard” dataset. The interrater agreement was calculated using Fleiss’ Kappa statistics. The Kappa coefficient ( $\kappa$ ) for sentence scoring is 0.782, showing good reliability of the interrater agreement. This dataset contains 90 positive, 87 negative, and 23 neutral sentences. More detail about the dataset is given in Table 5.

The second dataset, Dataset B, is the sentence polarity dataset v1.0<sup>3</sup> 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 in accord with the purpose of our first experiment, namely, to answer how efficiently the system can assess sentiments at sentence level.

The evaluation to assess the accuracy of paragraph (or document) level sentiment recognition is performed using Datasets C and D. We consider a paragraph (or document) as a set of sentences and the sentiment for a paragraph (or document) is currently assessed by considering the average score obtained from the scores of the sentences of the pertaining paragraph (or document). Dataset C is the polarity dataset V2.0 introduced in Pang and Lee (2004), which consists of 1000 positive and 1000 negative review documents. This dataset has become the *de facto* standard dataset for

**TABLE 5** Input Datasets

Dataset	Data type	Data attributes	Data source
Dataset A	Sentence	Data collected from various domains. 90 Positive, 87 Negative, and 23 Neutral sentences. More specifically the contexts and sentences are: E-mail: 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	Sentence	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 single snippet (usually containing roughly one single sentence); all snippets are down-cased.	Sentence polarity dataset v1.0. Introduced in Pang and Lee at ACL 2005. Can be found in at this source. <sup>4</sup>
Dataset C	Paragraph	Movie Review: 1000 positive and 1000 negative processed reviews.	Polarity dataset v2.0. Introduced in Pang and Lee at ACL 2004. can be found at this source. <sup>3</sup>
Dataset D	Paragraph	Product Review: 50 positive, 50 negative reviews about different products, including computers, mp3 players, mobile phones, cars, vacuum cleaners, TVs, and washing machines taken from epinions.com.	The authors collected this data from the website www.epinions.com

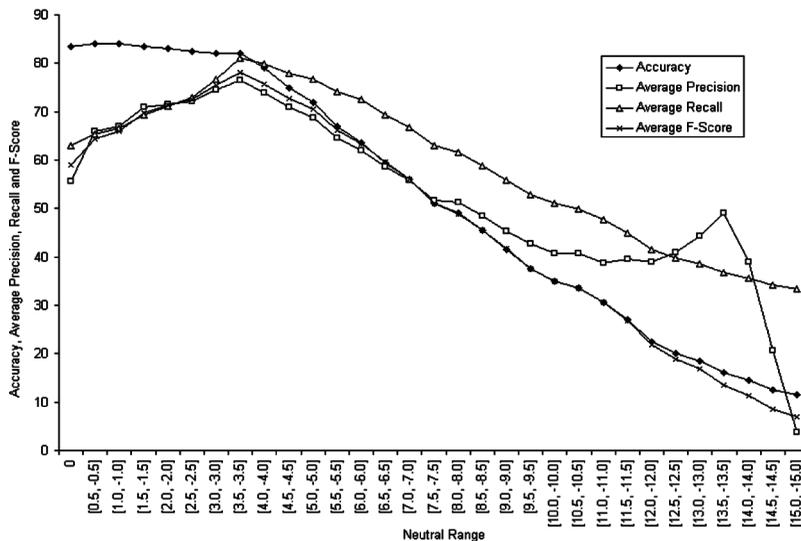
sentiment classification and has been used in over 15 research papers. Since movie reviews are known to be difficult to classify (Turney 2002; Turney and Littman 2003), we are motivated to test the performance of our system with such data.

Dataset D is a set of 100 reviews taken from epinions.com. This dataset contains 50 positive and 50 negative reviews. The reviews were collected from a variety of product reviews, including reviews on computers, mp3 players, mobile phones, cars, vacuum cleaners, TVs, and washing machines. Reviews at epinions.com are rated with a 5-star system where 1 is the lowest score and 5 is the highest score. Reviews where the product gets 1 or 2 stars are considered to be negative, reviews with 4 or 5 stars are considered to be positive. The purpose of using Dataset D is to measure the accuracy of the system in assessing the sentiment from product reviews. Both of the datasets (i.e., Datasets C and D) contain more than four sentences in each review. A summary of our “gold standard” datasets is given in Table 5.

## Sentence Level Comparisons

### 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 two other 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 Figure 4. According to Figure 4, the system achieves the highest accuracy (84%) for the ranges  $\pm 0.5$  and  $\pm 1.0$ , but it shows the highest average recall (81.04%), precision (76.49%), and F-score (78%) for the neutral range  $\pm 3.5$ . 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.



**FIGURE 4** 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.

**TABLE 6** 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

### *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 6 summarizes the accuracy of different approaches including ours for this dataset.

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 Pang and Lee (2005) and Eriksson (2006). 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 a precision of 92.07% and 91.01% for classifying positive and negative sentences, respectively.

## **Paragraph Level Comparisons**

### *Comparing to Machine-Learning Approaches*

Dataset C has been tested by comparing various approaches, including approaches based on machine-learning algorithms. While we built our system mainly to assess sentence level sentiment, we carried out this experiment in order to investigate the performance of the system when processing chunks of sentences (i.e., a paragraph or document). The method to obtain a score for text chunks is straightforward. We obtain its score by averaging over the scores of individual (positively and negatively scored) sentences. For example, the following excerpt is taken from one of the positive movie reviews found in Dataset C (only three subjective sentences are given for space limitation).

“If you want some hearty laughs, then rat race is the movie for you. This unpretentious little comedy, which sneaks into theaters today with very

**TABLE 7** The summary of several Systems that Experimented with Dataset C

Machine Learning Systems					
Experimental result reported in (Vincent, Dasgupta and Arifin 2006)	Accuracy				
Adding bigrams and trigrams	89.2				
Adding dependency relations	89.0				
Adding polarity info of adjectives	90.4				
Discarding objective materials	90.5				
		Accuracy			
Experimental results reported in (Mullen and Collier 2004)	3 folds (%)	10 folds (%)			
Pang et al. (2002)	82.9	NA			
Turney Values only	68.4	68.3			
Osgood only	56.2	56.4			
Turney Values and Osgood	69.0	68.7			
Unigrams	82.8	83.5			
Unigrams and Osgood	82.8	83.5			
Unigrams and Turney	83.2	85.1			
Unigrams, Turney, Osgood	82.8	85.1			
Lemmas	84.1	85.7			
Lemmas and Osgood	83.1	84.7			
Lemmas and Turney	84.2	84.9			
Lemmas, Turney, Osgood	83.8	84.5			
Hybrid SVM (Turney and Lemmas)	84.4	86.0			
Hybrid SVM (Turney/Osgood and Lemmas)	84.6	86.0			
Nonmachine Learning Systems					
Experimental result reported in (Kennedy and Inkpen 2006)	Accuracy (A) (%), Precision (P) (%), Recall (R) (%) for Positive and Negative Class				
Basic: GI	A = 59.5; P = 57.8,69.8; R = 82.8, 36.1;				
Basic: GI & CTRW & Adj	A = 65; P = 64.5,69.6; R = 73.3, 56.6;				
Basic: GI & SO-PMI 1	A = 57.7; P = 87.9,54.6; R = 18.8, 96.6				
Basic: GI & SO-PMI 2	A = 63.2; P = 61.1,73.5; R = 82.5,43.8				
Improved: GI	A = 62.7; P = 59.8,71.1; R = 81.7, 43.6				
Improved: GI & CTRW & Adj	A = 66.7; P = 65.8,70; R = 73.4,60.1				
Improved: GI & SO-PMI 1	A = 58.4; P = 87.3,55.1; R = 20,96.8				
Improved: GI & SO-PMI 2	A = 65.1; P = 61.9,73.9; R = 81.6,48.6				
Our Approach [Nonmachine Learning System]					
	Class	Accuracy (%)	Precision (%)	Recall (%)	F-score (%)
Our System (i.e., SenseNet)	Positive	85.5	87.78	79.7	83.54
	Negative		83.60	91.3	87.28

little hype, will have you bouncing in your theater seat. And while the film fits neatly into the low-brow, slapstick school of comedy, one refreshing aspect is its lack of mean-spiritedness.”

The system obtained +4.90 as the score for the above paragraph, whereby the three sentences received the scores 11.11, -7.72, and 11.32, respectively. Performance results for both machine-learning and nonmachine-learning-based approaches are reported in Table 7.

As already pointed out by Turney (2002), we notice that the movie review data contains a large portion of “objective” data (i.e., the text that describes about the plot of the movie), which cause noise in the analysis. As a review can contain both subjective and objective phrases, review identification can be viewed as an instance of the broader task of identifying which sentences in a document are factual/objective, and which are opinionated/subjective. There have been attempts on tackling this so-called document-level subjectivity classification task, with very encouraging results (see Yu and Hatzivassiloglou [2003] and Wiebe, Wilson, Bruce, Bell, and Martin [2004] for details). Our system outperformed the nonmachine-learning approaches, and achieved almost the same result as hybrid SVM (Turney/Osgood and lemmas) approach. The approach “Discarding objective materials” achieved the best performance using this dataset. However, in that experiment, first, the objective sentences are detected from the input review, and then classification is done based on the auto-detected subjective sentences of the review. In our opinion, if the objective sentences could be omitted, the performance of our system would increase but at present we have not considered preprocessing in order to filter the objective sentences.

### *Comparing to Online Rating as Gold Standard*

Dataset D is a set of 100 reviews taken from www.epinions.com. Table 8 summarizes the experimental result using this dataset.

**TABLE 8** Experimental Result Using the Dataset C

Review Data Genre	Class/Sample Size	Accuracy	Precision	Recall	F-score
Computer	Positive/8	78.57	85.71	75.00	80.00
	Negative/6		71.43	83.33	76.92
mp3 Player	Positive/6	72.73	66.67	66.67	66.67
	Negative/5		80.00	80.00	80.00
Mobile phone	Positive/10	85.00	80.00	80.00	80.00
	Negative/10		90.00	90.00	90.00
Automobile	Positive/8	83.33	77.78	87.50	82.35
	Negative/10		88.89	80.00	84.21
Vacuum Cleaner	Positive/4	87.50	100.00	75.00	85.71
	Negative/4		80.00	100.00	88.89
TV	Positive/9	83.33	80.00	88.89	84.21
	Negative/9		87.50	77.78	82.35
Washing Machine	Positive/5	81.82	80.00	80.00	80.00
	Negative/6		83.33	83.33	83.33
Average	Positive/50	81.75	81.45	79.01	79.85
	Negative/50		83.0	84.92	83.67

The result shows that the system's performance for product reviews (i.e., 81.75% accuracy) and movie reviews (i.e., 85.5% accuracy) does not vary significantly. In the approach (Turney, 2002), on the other hand, the movie review data achieved lower accuracy than product review data.

### Evaluating Individual Components of Our System

In order to evaluate individual components of our system, we prepared different versions (or models) such that some rules are either present or absent. Since our system implements several rules to deal with adjectives, adverbs, negations, conditions, and dependencies to get the contextual valence of the semantic verb frame(s) triplets (discussed in contextual valence assessment), different versions of our system are realized by either considering or not considering the respective rules, as follows.

- a. The “no ADJ” version of the system does not consider the rules that handle the adjectives in contextual valence assessment. Thus for the sentence, “*in a time when so many movies are timid and weak, American history x manages to make a compelling argument for racism without advocating it any way,*”, the “no ADJ” version does not consider the adjectives “timid”, “weak”, “compelling” while scoring this sentence. It hence outputs a lower score (i.e., 7.04) than the complete system (i.e., 10.81). In some cases (e.g., “*I would scale down the movie for its very poor visual effect.*”), this version outputs complete different sentiment than that of the original system.
- b. The “no ADV” version of the system does not consider the rules dealing with adverbs. Thus, for an example positive review sentence, “*Animated film ‘Monster House’ rarely receive critical raves.*” This model outputs a negative sentiment ( - 12.47) as it does not consider the adverb “rarely”.
- c. The “no ADJ & no ADV” model is the combination of the two models above. Hence, we expect to receive lower recall and F-scores for this system based on the hypothesis that both adjective and adverb are important linguistic components to assess sentiment from the text. Hence, the hypothesis is supported by the obtained result given in Table 9. We notice that this model of the system received lower accuracy and average F-scores for all the datasets than that of the two models above.
- d. The “no NEG & no CND” version of the system does not consider negation and conditionality while calculating contextual valence. So, for a sentence present in Dataset B, “*It’s a shame that his full talents were not used to full effect here,*” the system assesses the first triplet as a negative one but the second one is assessed as positive for not considering the

**TABLE 9** Experimenting with Different Models of the System Using All the Datasets

Model	System Performance Measures						
	Datasets	Class	Accuracy (%)	Precision (%)	Recall (%)	F-score (%)	
no ADJ	Dataset A	Positive	61.5	68.18	66.67	67.42	
		Negative		61.45	58.62	60.00	
		Neutral		41.38	52.17	46.15	
	Dataset B	Positive	75.26	77.58	73.21	75.33	
		Negative		73.18	77.30	75.19	
	Dataset C	Positive	67.05	68.42	61.30	64.66	
		Negative		65.94	72.80	69.20	
	Dataset D	Positive	70.00	69.81	74.00	71.84	
		Negative		70.21	66.00	68.04	
	no ADV	Dataset A	Positive	76	83.53	78.89	81.14
			Negative		71.26	71.26	71.26
			Neutral		67.86	82.61	74.51
Dataset B		Positive	80.10	78.74	79.40	79.07	
		Negative		81.48	80.79	81.13	
Dataset C		Positive	76.70	78.59	70.10	74.10	
		Negative		75.18	83.30	79.03	
Dataset D		Positive	60.00	53.45	62.00	57.41	
		Negative		69.05	58.00	63.04	
no ADJ & no ADV		Dataset A	Positive	55.5	59.34	60.00	59.67
			Negative		52.38	50.57	51.46
			Neutral		52.00	56.52	54.17
	Dataset B	Positive	63.82	65.39	62.00	63.65	
		Negative		62.41	65.65	63.99	
	Dataset C	Positive	49.05	36.00	33.30	34.60	
		Negative		60.28	64.80	62.46	
	Dataset D	Positive	48.00	43.75	56.00	49.12	
		Negative		55.56	40.00	46.51	
	no NEG & no CND	Dataset A	Positive	73.5	75.79	80.00	77.84
			Negative		75.00	68.97	71.86
			Neutral		60.00	65.22	62.50
Dataset B		Positive	81.98	84.33	84.47	84.40	
		Negative		79.63	79.50	79.56	
Dataset C		Positive	78.30	79.64	75.50	77.52	
		Negative		77.09	81.10	79.04	
Dataset D		Positive	69.00	68.75	66.00	67.35	
		Negative		69.23	72.00	70.59	
no Dependency		Dataset A	Positive	55	61.45	56.67	58.96
			Negative		54.76	52.87	53.80
			Neutral		39.39	56.52	46.43
	Dataset B	Positive	60.17	62.70	58.99	60.79	
		Negative		57.92	61.34	59.58	
	Dataset C	Positive	53.05	40.77	35.80	38.13	
		Negative		62.66	70.30	66.26	
	Dataset D	Positive	56.00	51.79	58.00	54.72	
		Negative		61.36	54.00	57.45	

negation. Thus, finally a positive valence is set for the sentence according to the rule for “not\_to\_dependency” triplets where a negative triplet precedes a positive triplet. Thus, this model signals this sentence

- as a positive sentence, although the sentence indicates a negative sentiment.
- e. The “no Dependency” version of the system does not consider the rules (as discussed in Sentiment Assessment) processing the dependency relationships between the triplets. Instead, it considers the average score of the triplets obtained from an input sentence. For the sentence present in Dataset B, “*The producers of this crow were either too dim to realize their story was doomed to be a hollow rehash, or too cynical to figure their audience would know the difference,*” this model did not apply the rules that process dependency and thus misclassified it as a positive sentence, whereas it is a part of a negative review and the original system scored it  $-9.61$  to classify it as a negative one.

The outcomes of the experimental results employing all the datasets by the models of the system discussed above are summarized in Table 9.

We observe that the “no ADJ & no ADV” and “no Dependency” model shows the worst performance over all the datasets. This reinforces our belief that adjectives, adverbs, as well as the relationships among the semantic verb frames of a sentence are very important linguistic clues to assess the sentiment of text.

### Comparison to the *EmpathyBuddy* System of Liu et al. (2003)

Although the system *EmpathyBuddy* (Liu et al. 2003) does not directly assess sentiment of text (as our system does), it is known for its outstanding performance in analyzing emotion from text of a 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. (2003), “*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%),

**TABLE 10** Accuracy Comparison Metrics Between Liu's System and Ours

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

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 10 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  $\pm 3.5$  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. (2003). 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.

## DISCUSSION

The goal of the previous section was to compare our rule-based approach to other methods for sensing sentiment of text. For this purpose, we performed experiments with four datasets. The results of the experiments indicate that our approach has an improving effect with regard to the classification of reviews. We could show that using our approach, accuracy, and recall for Dataset B are improved over other methods (i.e., gain of 7.44% on accuracy). Table 7 shows that our approach attains a gain of 18.8 percentage points (from 66.7% to 85.5%) over nonmachine-learning approaches, when applied on movie review data (i.e., Dataset C). In general, our approach also shows better performance than machine-learning approaches, with the exception of Hybrid SVM, which is 0.50 percentage points over our approach (see Table 7).

Since Dataset A and D are our original datasets, we could not compare the results to other methods. The specialty of Dataset A is that it has three types of sentences including neutral sentences. The experiment with this dataset revealed that if the valence range is  $\pm 3.5$  to signal neutrality, the average recall and F-score are maximized to 81.04% and 76.49%, respectively. For Dataset D, we achieved an accuracy of 81.75% with a recall of 79.01% and 84.92% for positive and negative sentence classes, respectively.

Movie reviews usually contain many sentences with “objective” information about the characters or the plot of the movie. Although these sentences are “objective” (in the sense of not being subjective), they may contain positive and negative terms. This is also true of movie titles, for example, “Ghost,” “Pirates of the Caribbean: Dead Man’s Chest,” “Star Wars,” “Mission: Impossible,” “Die Another Day,” etc. These are very positively reviewed movies; however, their titles contain some negative terms. Repeating of such titles of the film in the review would make the review seem more negative (or positive for negative reviews for the titles with positive terms). Similar problems might exist for product reviews, maybe to a lesser extent (as pointed out by Turney [2002]). In order to validate this claim, we experimented with both types of data: movie review and product review. Datasets B and C are movie review data, and Dataset D contains product reviews. The percentage differences between the accuracy and average recall obtained for Datasets C and D are 3.75% and 3.54%, respectively, which indicates that the system shows better accuracy for movie review data than product review data. Hence, in our opinion, although there are objective sentences in the input text, and the system treats those objective sentences as if they were subjective, the average score of all the sentences of the whole input text is similar to the “gold standard” ranking.

Our system is robust in the sense that it can tackle the case where a negative term containing a movie title for a positive review or vice versa may produce wrong outputs by keyword spotting or machine-learning approaches. Since our system works on the basis of semantic structure of the sentence it considers the name as the subject or object of the sentence having attributes and emphasis on the scoring of a verb to which it is associated. Thus for the input sentence, “*at the end of the film Pirates of the Caribbean: Dead Man’s Chest, I was involved in the characters, and I was satisfied with the outcome,*” the system found two positive verbs, namely, “involve” and “satisfy with” associated with the object “characters” and “outcome” where the object “characters” is having the attributes “film,” and “Pirates of the Caribbean: Dead Man’s Chest” which finally assign a positive contextual valence to the object “characters” according to our algorithm. Thus, our system output for this sentence is +8.453, that is, a positive sentence. On the other hand, keyword spotting-based machine-learning and nonmachine-based approaches, such as Polanyi and Zaenen (2004) and Kennedy and Inkpen (2006) will produce the wrong output for such cases.

Like the work of Liu et al. (2003), our approach to sensing affective information from text relies on common sense knowledge, which contributes to their robustness. Textual information (e.g., nouns) is mapped to concepts that are derived from a large-scale, real-world knowledge base of common sense knowledge. The concepts usually have inherent affective connotation, such as “positive” or “negative,” “happy” or “sad” etc. Hence

for the input “*Mary was invited to Jack’s party. She wondered if he would like a kite. She went and shook her piggy bank. It made no sound,*” humans apply common sense to draw the following inferences: Gift is related with a party. Kite may be a gift item. Money is essential to buy a gift. If there is no coin in a piggy bank, no clattering sound is produced. No money and no gift make someone discouraged for the party. In this case, the common sense model should relate “party” (i.e., positive event) to a “gift” concept (i.e., positive concept) and finally obtained a scenario mapped to negative concept “no money.” Relating real-world scenarios to concepts and concepts to emotional affinity works well when the sentences are semantically simple and descriptive. But common sense-based approaches may fail for the sentences, “*You will hardly get a bad shot with this camera,*” and “*the three simple words you need to know in order to make your choice about owning your own iPod nano are: It’s Sexy. It’s Sleek. It’s Small.*” They may fail because, first, they do not consider the semantic structure of the sentence and second, they may not have knowledge about concepts such as “iPod” to assess emotional affinity.

Our approach overcomes such problems because we consider the semantic structure of the sentence and then assign the contextual valence based on the assessment of the semantic verb frame(s). In fact, we also have incorporated the common sense knowledge in terms of assigning prior valence values to words and implementing the rules to process the linguistic components for valence assignment. Moreover, we employ online resources to assess positive/negative opinions about new concepts (e.g., iPod), which might not (yet) or never be part of the common sense knowledge base. In our opinion, our approach is robust and can be thought as an improvement over the common sense-based approach because common sense approach maps a description to a collection of concepts and then concepts to their affective nature of everyday situations to classify sentences into “basic” emotion categories (i.e., either negative or positive), whereas our approach employs common sense knowledge base to assign words either a negative or positive score, considers the semantic structure of the sentence, and apply rules to assign the contextual valence of the so-called concepts (i.e., semantic verb frame) and their associated relationships obtained from the sentence.

We are using different linguistic resources in order to assign prior valence to words (see The Knowledge base). Our notion of “prior valence” is sometimes called “semantic orientation” (SO) in the literature (Hatzivassiloglou and McKeown 2002). We are aware of the procedures mentioned in Turney and Littman (2003) and Grefenstette et al. (2004), which employed a hit result (of search engines) method to assign different semantic axes (i.e., positive or negative, excellent or bad, etc.) to words. Due to some limitations of the SO approach mentioned in Turney (2002) and Turney and Littman (2003), we motivate a new approach that incorporates

(1) WorldNet-based manual scoring for verbs and adjectives; (2) common sense knowledge to score nouns; and (3) Internet-based resources to score named entities. In our future work we plan to evaluate the scores obtained by our approach with respect to other approaches (e.g., SentiWordNet).

Our system builds the computational model of the input sentence after the output of the language parser. We have noticed several problems with the language parser. For example, for the input, "*pretty cool movie though,*" the sentence/expression does not contain a verb, and hence the computation model (i.e., triplet) cannot be formed. In such cases, we scored it by calculating the context valence considering the adjectives and nouns (i.e., similar to the keyword spotting-based approach). We observed that for the malformed or incomplete or too fragmented sentences, the semantic parser sometimes outputs erroneous triplets in terms of identifying the interdependencies between the triplets. For example, a sample review sentence, "*There's more, I suppose, but it's not worth it; the acting is bland, neither arsenic nor gravy; the music disposable; the camera work turgid,*" formed erroneous triplets because of possible missing verb in this part (i.e., *the music disposable*) and the parser considered those linguistic components as the attributes of the last well-formed triplet and the contextual valence is calculated thereby. Thus malformation of a triplet might be one of the sources of our errors.

We also observed that sometimes our approach of automatically assigning a new valence value to a nonscored new word outputs erroneous valence, which causes wrong classification of sentence. For example, for the input sentence, "*Everything in the movie is so forced, so unauthentic that anyone with an i.q. over 80 will know they wasted their money on an unfulfilled desire,*" our automatic approach assigned a positive score (i.e., 1.363) for the adjective "unauthentic" which made the evaluation of the first triplet (i.e., [['Subject Name:', 'sb/sth', 'Subject Type:', ', ', 'Subject Attrib:', []], ['Action Name:', 'force', 'Action Status:', 'Past Participle', 'Action Attrib:', ['ADV: so', 'passive', 'dependency: that']], ['Object Name:', 'everything', 'Object Type:', 'Object', 'Object Attrib:', ['Determiner: the', 'N NOM SG: movie', 'ADV: so', 'A ABS: unauthentic']]]) as a positive one, although it is negative. In our experience, the major reason for generating wrong outputs by the system is caused by this process of automatic scoring of new words. Hence, we plan to revise our method of assigning prior valence values for new words by investigating other approaches like affect control theory (Heise 2007), which assigns different scores (i.e., evaluation, potency, and activity) for a word based on different social settings (e.g., culture, situations, etc.).

## CONCLUSION

The new discipline coined as "affective computing" (Picard 1997) investigates the basics of human emotion and emphasizes both

the physiological and cognitive aspects of emotion. The affective computing community developed several mechanisms for emotion sensing, including the processing of various physiological signals obtained from wearable sensors. A complementary research direction originating in natural language processing puts an emphasis on emotion sensing from text. We believe that text is an important modality for computer-human interaction, and sensing of textual affective information can significantly contribute to the success of affective user interfaces and intelligent machines. For the task of emotion sensing, text can also complement other modalities like speech or gesture (as reported in Russell, Bachorowski, and Fernandez-Dols [2003]), and thus increase the robustness of emotion recognition.

The system described in this article 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 common sense 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, this 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 plan to extend the sentiment recognition system into a full-fledged emotional recognition system, which may classify named emotions rather than positive or negative sentiments. We plan to follow the OCC emotion model (Ortony, Clore, and Collins 1988) by applying different linguistic tools and heuristics to sense a rich set of affective information from the text. We also intend to take into account user-specific preferences (e.g., personal opinions about particular entities) that might help the system to analyze subjective statements in a personalized manner.

## REFERENCES

- Carenini, G., R. T. Ng, and E. Zwart. 2005. Extracting knowledge from evaluative text, In: *Proceedings of the 3rd International Conference on Knowledge Capture Alberta*, Canada, pp. 11–18.
- Cesarano, C., B. Dorr, A. Picariello, D. Reforgiato, A. Sagoff, and V. S. Subrahmanian. 2006. OASYS: An opinion analysis system, In: *Proceedings of AAAI-2006 Spring Symposium on Computational Approaches to Analyzing Weblogs*, Boston, MA, pp. 21–26.

- English Vocabulary. 2006. English Club. <http://www.englishclub.com/vocabulary>. Last accessed December, 2006.
- Eriksson, B. 2006. Sentiment classification of movie reviews using linguistic parsing. [http://pages.cs.wisc.edu/~apirak/cs/cs838/eriksson\\_final.pdf](http://pages.cs.wisc.edu/~apirak/cs/cs838/eriksson_final.pdf). Last accessed October, 2007.
- Esuli, A. and F. Sebastiani. 2005. Determining the semantic orientation of terms through gloss analysis. In: *Proceedings of the ACM International Conference on Information and Knowledge Management (CIKM-2005)*, Bremen, Germany, pp. 617–624.
- Esuli, A. and F. Sebastiani. 2006. SentiWordNet: A publicly available lexical resource for opinion mining. In: *Proceedings of LREC-06, 5th Conference on Language Resources and Evaluation*, Genova, Italy, pp. 417–422.
- Fellbaum, C. (ed.). 1999. *WordNet: An Electronic Lexical Databases*. Cambridge, MA: MIT Press.
- Fitriane, S. and L. J. M. Rothkrantz. 2006. Constructing knowledge for automated text-based emotion expressions. In: *Proceedings of the International Conference on Computer Systems and Technologies, (CompSysTech-06)*, Vol. 6, Veliko Tarnovo, Bulgaria, pp. 1–6.
- Fleiss, J. L. 1971. Measuring nominal scale agreement among many raters. *Psychological Bulletin* 76(5): 378–382.
- Gamon, M. 2004. Sentiment classification on customer feedback data: noisy data, large feature vectors, and the role of linguistic analysis. In: *Proceedings of the 20th International Conference on Computational Linguistics*, Geneva, Switzerland, pp. 841–847.
- Grefenstette, G., Y. Qu, D. Evans, and J. Shanahan. 2004. Validating the coverage of lexical resources for affect analysis and automatically classifying new words along semantic axes. In: *Computing Attitude and Affect in Text: Theory and Applications*, eds. J. Shanahan, Y. Qu, and J. Wiebe, pp. 93–107. The Information Retrieval Series Vol. 20, Netherlands: Springer-Verlag.
- Hatzivassiloglou, V. and K. R. McKeown. 2002. Predicting the semantic orientation of adjectives. In: *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and the 8th Conference of the European Chapter of the ACL*. Madrid, Spain, pp. 174–181.
- Heise, D. R. 2007. *Expressive Order Confirming Sentiments in Social Actions*. New York: Springer.
- Hu, M. and B. Liu. 2004. Mining and summarizing customer reviews. In: *Proceedings of the International Conference on Knowledge Discovery and Data Mining, (KDD-04)*, Seattle, WA, pp. 168–177.
- Johnson, C. R., C. J. Fillmore, M. R. L. Petruck, C. F. Baker, M. Ellsworth, J. Ruppenhofer, and E. J. Wood, eds. 2006. *FrameNet II: Theory and Practice*. <http://framenet.icsi.berkeley.edu/book/book.pdf>. Last accessed December, 2006.
- Kamps, J. and M. Marx. 2002. Words with attitude. In: *Proceedings of the First International Conference on Global WordNet*, Mysore, India.
- Kennedy, A. and D. Inkpen. 2006. Sentiment classification of movie reviews using contextual valence shifter. *Computational Intelligence*, 22(2):110–125.
- Kim, S. M. and E. Hovy. 2006. Identifying and analyzing judgment opinions. In: *Proceedings of the HLT-NAACL International Conference*, New York, pp. 200–207.
- Kim, S. M. and E. Hovy. 2005. Automatic detection of opinion bearing words and sentences. In: *Proceedings of the International Joint Conference on Natural Language Processing, (IJCNLP-05)*, Jeju Island, Korea, pp. 61–66.
- Knobloch, S., G. Patzig, A. Mende, and M. Hastall. 2004. Affective news Effects of discourse structure in narratives on suspense, curiosity, and enjoyment while reading news and novels. *Communication Research* 31(3):259–287.
- Koppel, M. and I. Shtirimberg. 2004. Good news or bad news? Let the market decide. In: *Proceedings of the AAAI Spring Symposium on Exploring Attitude and Affect in Text*, Palo Alto, CA, pp. 86–88.
- Liu, H., H. Lieberman, and T. Selker. 2003. A model of textual affect sensing using real-world knowledge. In: *Proceedings of the International Conference on Intelligent User Interface (IUI-03)*, Miami, FL, pp. 125–132.
- Liu, H. and P. Singh. 2004. ConceptNet: A practical commonsense reasoning toolkit. *BT Technology Journal* 22(4): 211–226.
- Machinese Syntax. 2007. Connexor. <http://www.connexor.eu/technology/machinese/machinese-syntax>. Last accessed January, 2008.
- Mihalcea, R. and H. Liu. 2006. A corpus-based approach to finding happiness. In: *Proceedings of the AAAI Spring Symposium on Computational Approaches for Analysis of Weblogs*, California, Stanford.
- Mullen, T. and N. Collier. 2004. Sentiment analysis using support vector machines with diverse information sources. In: *Proceedings of the International Conference on Empirical Methods in Natural Language Processing (EMNLP-2004)*, Barcelona, Spain, pp. 412–418.

- My Yahoo! 2007. My Yahoo! <http://my.yahoo.com>. Last accessed Feb, 2008.
- Nasukawa, T. and J. Yi. 2003. Sentiment analysis: Capturing favorability using natural language processing. In: *Proceedings of the 2nd International Conference on Knowledge Capture (K-CAP'03)*, Florida, Sanibel, pp. 70–77.
- Opinmind. 2006. Discovering Bloggers. <http://www.opinmind.com>. Last accessed November, 2007.
- Ortony, A., G. L. Clore, and A. Collins. 1988. *The Cognitive Structure of Emotions*. New York: Cambridge University Press.
- Pang, B. and L. Lee. 2005. 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, Ann Arbor, pp. 115–124.
- Pang, B. and L. Lee. 2004. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In: *Proceedings of the 42nd Annual Meeting of the Association of Computational Linguistics (ACL-04)*, Barcelona, Spain, pp. 271–278.
- Pang, B., L. Lee, and S. Vaithyanathan. 2002. Thumbs up? Sentiment classification using machine learning techniques. In: *Proceedings of the International Conference on Empirical Methods in Natural Language Processing (EMNLP-02)*, Philadelphia, PA, pp. 79–86.
- Pennebaker, J. W., M. R. Mehl, and K. G. Niederhoffer. 2003. Psychological aspects of natural language use: Our words, our selves. *Annual Review of Psychology*, 54:547–577.
- Pennebaker, J. W., M. E. Francis, and R. J. Booth. 2001. *Linguistic Inquiry and Word Count: LIWC* (2nd ed.). Mahwah, NJ: Erlbaum.
- Picard, R. W. 1997. *Affective Computing*. Cambridge, MA: The MIT Press.
- Polanyi, L. and A. Zaenen. 2004. Contextual valence shifters. In: *Computing Attitude and Affect in Text: Theory and Applications*, eds. J. Shanahan, Y. Qu, and J. Wiebe, pp. 1–10. The Information Retrieval Series Vol. 20. Netherlands: Springer Verlag.
- Riloff, E., J. Wiebe, and T. Wilson. 2003. Learning subjective nouns using extraction pattern bootstrapping. In: *Proceedings of the 7th International Conference on Natural Language Learning (CoNLL-03)*, Edmonton, Canada, pp. 25–32.
- Rosis, F. and F. Grasso. 2000. Affective natural language generation. *Affective Interactions, Towards a New Generation of Computer Interfaces*, ed. A. Paiva, pp. 204–218. New York: Springer-Verlag.
- Russell, J. A., J. A. Bachorowski, and J. M. Fernandez-Dols. 2003. Facial and vocal expressions of emotion. *Annual Review of Psychology* 54:329–349.
- Sebastiani, F. 2002. Machine learning in automated text categorization. *ACM Computing Surveys* 34(1): 1–47. Last accessed November, 2006.
- Sentiment! 2005. Corpora Software: <http://www.corporasoftware.com>.
- Shaikh, M. A. M., H. Prendinger, and M. Ishizuka. 2006a. A cognitively based approach to affect sensing from text. In: *Proceedings of the 10th International Conference on Intelligent User Interface (IUI-06)*, Sydney, Australia, pp. 349–351.
- Shaikh, M. A. M., M. T. Islam, and M. Ishizuka. 2006b. 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, pp. 133–138.
- Shaikh, M. A. M., H. Prendinger, and M. Ishizuka. 2007a. SenseNet: A linguistic tool to visualize numerical-valence based sentiment of textual data. In: *Proceedings of the International Conference on Natural Language Processing (ICON-07)*, Hyderabad, India, pp. 147–152.
- Shaikh, M. A. M., H. Prendinger, and M. Ishizuka. 2007b. 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, pp. 191–202.
- Stock, O. and C. Strapparava. 2003. Getting serious about the development of computational humor. In: *Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI-03)*, Acapulco, Mexico, pp. 59–64.
- Subasic, P. and A. Huettner. 2001. Affect analysis of text using fuzzy semantic typing. *IEEE Transactions on Fuzzy Systems* 9(4):483–496.
- Turney, P. 2002. Thumbs up or thumbs down? semantic orientation applied to unsupervised classification of reviews. In: *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics (ACL-02)*, Pennsylvania, Philadelphia, pp. 417–424.

- Turney, P. D. and M. L. Littman. 2003. Measuring praise and criticism: Inference of semantic orientation from association. *ACM Transactions on Information Systems (TOIS)* 21(4):315–346.
- Valitutti, A., C. Strapparava, and O. Stock. 2004. Developing affective lexical resources. *PsychNology Journal* 2(1):61–83.
- Vincent, N., S. Dasgupta, and S. M. N. Arifin. 2006. Examining the role of linguistic knowledge sources in the automatic identification and classification of reviews. In: *Proceedings of the COLING/ACL*, Sydney, Australia, pp. 611–618.
- Wiebe, J. 2000. Learning subjective adjectives from corpora. In: *Proceedings of the 12th International Conference on Innovative Applications of Artificial Intelligence (IAAI-00)*, Texas, Austin, pp. 735–740.
- Wiebe, J., R. Bruce, M. Bell, M. Martin, and T. Wilson. 2001. A corpus study of evaluative and speculative language. In: *Proceedings of the 2nd SIGdial Workshop on Discourse and Dialogue* (Vol. 16), Aalborg, Denmark, pp. 1–10.
- Wiebe, J. M., T. Wilson, R. Bruce, M. Bell, and M. Martin. 2004. Learning subjective language. *Computational Linguistics* 30(3):277–308.
- Wiebe, J., T. Wilson, and C. Cardie. 2005. Annotating expressions of opinions and emotions in language. *Language Resources and Evaluation* 39(2–3):165–210.
- Wiebe, J. and R. Mihalcea. 2006. Word sense and subjectivity. In: *Proceedings of the Association for Computational Linguistics Conference (ACL-06)*, Sydney, Australia, pp. 1065–1072.
- Wilson, T., J. Wiebe, and P. Hoffmann, 2005. Recognizing contextual polarity in phrase-level sentiment analysis. In: *Proceedings of the International Conference on Human Language Technology and Empirical Methods in Natural Language Processing (HLT/EMNLP-05)*, Vancouver, Canada, pp. 347–354.
- Yu, H. and V. Hatzivassiloglou. 2003. Towards answering opinion questions: Separating facts from opinions and identifying the polarity of opinion sentences. In: *Proceedings of the International Conference on Empirical Methods in Natural Language Processing (EMNLP-03)*, Sapporo, Japan, pp. 129–136.
- Zhe, X. and A. C. Boucouvalas. 2002. Text-to-emotion engine for real time internet communication. In: *Proceedings of the Third International Symposium on Communication Systems, Network and Digital Signal Processing (CSNDSP-02)*, Staffordshire, UK, pp 164–168.

## NOTES

1. The service is suspended recently but using the service 2300 named entities were scored previously.
2. <http://www.almasum.com/research/survey/> (one can login using “guest” as username).
3. Introduced in Pang and Lee at ACL 2005 at <http://www.cs.cornell.edu/People/pabo/movie-review-data/>.
4. <http://www.cs.cornell.edu/People/pabo/movie-review-data/>

## APPENDIX A

Pseudo code for assessing contextual valence considering adjective, adverb, negation, conditionality is given below.

**function** getValence (*P*)

outputValence = {}

Begin

for each  $S_i$  in *P* do // assume  $1 \leq i \leq n$

    tripletSet<sub>*i*</sub> = getSemantic Parsing ( $S_i$ )

        //the output of Semantic Parser is a set of triplets for each sentence.

    for each triplet  $T_j$  in tripletSet<sub>*i*</sub> do //we assume  $1 \leq j \leq m$ , *m* triplets

        actorValence = ContextualValenceAttrib (actorPriorValence, actor-Attributes)

```

actionValence = ContextualValenceAttrib (actionPriorValence, action
  Attributes)
objectValence = ContextualValenceAttrib (objectPriorValence, object
  Attributes)
actionObjectPairValence = setActionObjectPairVal (actionValence,
  objectValence)
tripletValence = setTripletValence (actorValence, actionObject
  PairValence)
tripletValence = handleNegationAndConditionality (tripletValence,
  Tj)
tripletDependency = if the token “dependency” is found then
  ‘true’ else ‘false’ is set
tripletDependencyType = ‘to_dependency’ or ‘not_to_dependency’
  based on tag
tripletResult Tj = {tripletValence, tripletDependency, triplet
  DependencyType}
loop until all triplets are processed
contextualValence = processTripletLevelContextualValence (tripletSeti)
m = sizeof(contextualValence)
sentimentScore = average (∑k=1m abs(contextualValencek))
valenceSign = getResultantValenceSign (contextualValence)
SentenceValencei = sentimentScore * valenceSign
outputValence = outputValence ∪ SentenceValencei
loop until all sentences are processed
valence = getParagraphValence (SentenceValence)
outputValence = valence ∪ {SentenceValence}
End
function processTripletLevelContextualValence (tripletSeti)
Begin
M = sizeOf(tripletSeti)
ContextualValence = [ ]
for k = 1 to M - 1 do
  R1: = tripletResultk
  R2: = tripletResultk+1
  if R1.tripletDependency = true and R1.tripletDependency! =
    “to_dependency”
    ContextualValencek = setContextualValence (R1.tripletValence,
      R2.tripletValence, “Not_To_Dependency”)
  else if R1.tripletDependency = false
    ContextualValencek = R1.tripletValence
end loop k
for k = 1 to M - 1 do

```

```

R1: = tripletResultk
R2: = tripletResultk+1
if R1.tripletDependency = true and R1.tripletDependency
Type = "to_dependency"
if ContextValencek+1! = null then
Begin
ContextualValencek = setContextualValence(R1.tripletValence,
ContextValencek+1, "To_Dependency")
ContextualValencek-1 = null
End
Else
ContextualValencek = setContextualValence(R1.tripletValence, R2.triplet
Valence, "To_Dependency")
end loop k
return ContextualValence
End

```

## APPENDIX B

The summary of experimental result for Dataset A using different range-to-signal neutrality of sentences is given below:

Neutral Range	Class	Accuracy	Precision	Recall	F-Score	Average Precision	Average Recall	Average F-Score
0	Positive	83.5	81.731	<b>94.444</b>	87.629	55.716	62.899	59.082
	Negative		85.412	<b>94.253</b>	89.617			
	Neutral		.001	0	0			
- 0.5 to 0.5	Positive	<b>84.0</b>	84.845	93.333	88.889	66.007	65.427	64.301
	Negative		88.172	94.253	<b>91.111</b>			
	Neutral		25	8.696	12.903			
- 1.0 to 1.0	Positive	<b>84.0</b>	86.598	93.333	89.840	66.870	66.493	65.998
	Negative		<b>89.011</b>	93.103	91.011			
	Neutral		25	13.043	17.143			
- 1.5 to 1.5	Positive	83.5	87.234	91.111	89.130	70.837	69.334	69.722
	Negative		87.778	90.805	89.266			
	Neutral		37.5	26.087	30.769			
- 2.0 to 2.0	Positive	83	90	90	90	71.537	71.096	71.288
	Negative		86.517	88.506	87.5			
	Neutral		38.095	34.783	36.364			
- 2.5 to 2.5	Positive	82.5	91.954	88.207	<b>90.395</b>	72.207	72.858	72.473
	Negative		86.207	86.207	86.207			
	Neutral		38.462	43.478	40.816			
- 3.0 to 3.0	Positive	82	91.765	86.667	89.143	74.587	76.765	75.409
	Negative		83.721	82.759	83.237			
	Neutral		48.276	60.870	53.846			

(Continued)

## APPENDIX B Continued

Neutral Range	Class	Accuracy	Precision	Recall	F-Score	Average Precision	Average Recall	Average F-Score
- 3.5 to 3.5	Positive	82	<b>92.771</b>	85.556	89.017	<b>76.486</b>	<b>81.042</b>	<b>78.002</b>
	Negative	82.143	79.310	80.702				
	Neutral	<b>54.545</b>	78.261	64.286				
- 4.0 to 4.0	Positive	79	90.123	81.111	85.380	73.988	79.861	75.607
	Negative		80.488	75.862	78.107			
	Neutral		51.351	82.609	63.333			
- 4.5 to 4.5	Positive	75	87.342	76.667	81.657	71.005	77.913	72.787
	Negative	74.390	70.115	72.189				
	Neutral	51.282	86.957	64.516				
- 5.0 to 5.0	Positive	72	87.013	74.444	80.240	68.716	76.706	70.507
	Negative	69.136	64.368	66.667				
	Neutral	50	91.304	<b>64.615</b>				
- 5.5 to 5.5	Positive	67	81.081	66.667	73.170	64.571	74.030	66.226
	Negative		65.823	59.770	62.651			
	Neutral		46.809	95.652	62.857			
- 6.0 to 6.0	Positive	63.5	80	62.222	70	61.953	72.465	63.331
	Negative		60.759	55.172	57.831			
	Neutral		45.098	<b>100</b>	62.162			
- 6.5 to 6.5	Positive	59.5	79.411	60	68.354	58.592	69.425	59.516
	Negative	54.545	48.276	51.220				
	Neutral	41.818	100	58.974				
- 7.0 to 7.0	Positive	56	76.923	55.556	64.516	55.752	66.794	56.029
	Negative	52	44.828	48.148				
	Neutral	38.333	100	55.423				
- 7.5 to 7.5	Positive	51	72.131	48.889	58.278	51.750	63.040	51.044
	Negative	49.296	40.230	44.304				
	Neutral	33.824	100	50.549				
- 8.0 to 8.0	Positive	49	75	46.668	57.534	51.164	61.533	48.927
	Negative	47.826	37.931	42.308				
	Neutral	30.667	100	46.939				
- 8.5 to 8.5	Positive	45.5	72.222	43.333	54.167	48.411	58.889	45.518
	Negative	44.615	33.333	38.158				
	Neutral	28.395	100	44.231				
- 9.0 to 9.0	Positive	41.5	70	38.889	50	45.373	55.875	41.717
	Negative	39.683	28.736	33.333				
	Neutral	26.437	100	41.818				
- 9.5 to 9.5	Positive	37.5	65.957	34.444	45.255	42.693	52.861	37.583
	Negative	38.889	24.138	29.787				
	Neutral	23.232	100	37.704				
- 10.0 to 10.0	Positive	35	62.222	31.111	41.481	40.709	50.983	35.052
	Negative	38	21.839	27.737				
	Neutral	21.905	100	35.938				
- 10.5 to 10.5	Positive	33.5	65.854	30	41.221	40.664	49.847	33.578
	Negative	35.417	19.540	25.185				
	Neutral	20.721	100	34.329				
- 11.0 to 11.0	Positive	30.5	64.865	26.667	37.795	38.670	47.586	30.521
	Negative	31.818	16.091	21.374				
	Neutral	19.328	100	32.394				

(Continued)

## APPENDIX B Continued

Neutral Range	Class	Accuracy	Precision	Recall	F-Score	Average Precision	Average Recall	Average F-Score
– 11.5 to 11.5	Positive	27	63.333	21.111	31.667	39.530	44.968	26.800
	Negative	38.710	13.793	20.339				
	Neutral	16.547	100	28.395				
– 12.0 to 12.0	Positive	22.5	52.174	13.333	21.239	38.941	41.609	21.829
	Negative	50	11.494	18.692				
	Neutral	14.650	100	25.556				
– 12.5 to 12.5	Positive	20	55.556	11.111	18.519	41.004	39.719	18.826
	Negative	53.846	8.046	14				
	Neutral	13.609	100	23.958				
– 13.0 to 13.0	Positive	18.5	53.333	8.889	15.238	44.356	38.595	16.951
	Negative		66.667	6.897	12.5			
	Neutral		13.068	100	23.116			
– 13.5 to 13.5	Positive	16	60	6.667	12	49.122	36.705	13.534
	Negative		75	3.448	6.594			
	Neutral		12.366	100	22.010			
– 14.0 to 14.0	Positive	14.5	71.429	5.556	10.309	38.956	35.568	11.376
	Negative		33.333	1.149	2.222			
	Neutral		12.105	100	21.596			
– 14.5 to 14.5	Positive	12.5	50	2.222	4.255	20.619	34.074	8.484
	Negative		.001	0	0			
	Neutral		11.856	100	21.198			
– 15.0 to 15.0	Positive	11.5	.001	0	0	3.834	33.333	6.876
	Negative		.001	0	0			
	Neutral		11.5	100	20.628			