

Assessing Sentiment of Text by Semantic Dependency and Contextual Valence Analysis

Mostafa Al Masum Shaikh¹, Helmut Prendinger², and Ishizuka Mitsuru¹

¹ Department of Information and Communication Engineering, University of Tokyo,
7-3-1 Hongo, Bunkyo-ku, 113-8656 Tokyo, Japan
mostafa@mi.ci.i.u-tokyo.ac.jp, ishizuka@i.u-tokyo.ac.jp

² Digital Contents and Media Sciences Research Division, National Institute of Informatics,
2-1-2 Hitotsubashi, Chiyoda-ku, 101-8430 Tokyo, Japan
helmut@nii.ac.jp

Abstract. 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, and an affect or emotion (e.g. happy, fearful, surprised etc.). We consider sentiment assessment and emotion sensing from text as two different problems, whereby sentiment assessment is a prior task to emotion sensing. This paper presents an approach to sentiment assessment, i.e. the recognition of negative or positive sense of a sentence. We perform semantic dependency analysis on the semantic verb frames of each sentence, and 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. Empirical results indicate that our system outperforms another state-of-the-art approach.

1 Introduction

Sensing of affective information would benefit the development of text based affective user interfaces because the words people use to express their feelings can be important clues to their mental, social, and physical state [15]. Examples of such applications are the affective text analyzer [2,5,18], the affective email-client [9], empathic chat, information and tutoring tools, computational humor [20], affective lexicon [22], affective information recognizer [12,7], and psycholinguistic analysis [6,15]. 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 relies on the semantic relationship between the structure of natural language and contextual valence of the words used in a given text. The scope of this paper, however, is limited to sentiment assessment.

There are four main factors that distinguish our work from others. First, we have integrated semantic processing of input text by dependency analysis on semantic verb-frame(s) of each sentence. Second, cognitive and commonsense knowledge resources have been utilized to assign a prior valence to a set of words, which

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 or relying on text corpora, we followed a rule-based approach to assess the valence of each semantic verb frame 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 sentiments from texts of any genre (e.g. movie or product review, news articles, blogs, etc.) at the sentence level.

2 Background and Related Work

Sentiment has been studied at three different levels: word, sentence, and document level. There are methods to estimate positive or negative sentiment of words [21,1], phrases and sentences [7,26], and documents [5]. Previous approaches for assessing sentiment from text are based on one or a combination of the following techniques: keyword spotting, lexical affinity [22,8], statistical methods [15], a dictionary of affective concepts and lexicon, commonsense knowledgebase [9], fuzzy logic [19], knowledge-base from facial expression [3], machine learning [7,25], domain specific classification [13], and valence assignment [16,26,18].

Some researchers proposed machine learning methods to identify words and phrases that signal subjectivity. For example, Wiebe and Mihalcea [23] stated that subjectivity is a property that can be associated with word senses, and hence word sense disambiguation can directly benefit subjectivity annotations. Turney [21] and Wiebe [24] concentrated on learning adjectives and adjectival phrases, whereas Wiebe et al. [25] focused on nouns. Riloff et al [17] extracted patterns for subjective expressions as well.

According to a linguistic survey [15], only 4% of the words used in written texts carry affective content. This finding shows that using affective lexicons is not sufficient to recognize affective information from text. It also indicates the difficulty of employing methods like machine learning, keyword spotting, and lexical affinity (see [9] for a detailed criticism). Statistical methods are well suited for psycholinguistic analysis (e.g. [15]) 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 pre-assigned 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.

A number of researchers have explored to automatically learn words and phrases with prior positive or negative valence (e.g., [6,21,26]). By contrast, we begin with a lexicon of words by calculating prior valences using WordNet [2] and ConceptNet [10], and assign the contextual valence [16] of phrases by applying a set of rules. Kim and Hovy [7], Hu and Liu [5], and Wilson et al [26] multiply or count the prior valences of opinion bearing words of the 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 [13] 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

of text has shown very promising results regarding sentiment analysis of product-reviews and blogs, but it requires special tuning of data in order to build category-specific classifiers for each text-domain (e.g. product review or movie review).

3 Our Approach

We propose a pipelined architecture with the following phases: Parse, Process, and Assess. 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-valences to the triplet(s). Finally, in the Assess phase an overall valence is assigned to each input sentence(s).

3.1 Semantic Parsing

For each input sentence the Semantic Parsing 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.). We first obtain XML-formatted syntactic and functional dependency information of each word of the input text using the Machinese Syntax parser [11] and this output constitutes the basis for further processing to generate the triplet(s). Since a triplet is initiated 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 and how” with a notion of semantic verb frame [2] 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.

Table 1. Triplet output of Semantic Parsing for the sentence given above

Senses processed by SenseNet	
Triplet 1	[[['Subject-Name:', 'raid', 'Subject-Type:', 'concept', 'Subject-Attrib:', ['A ABS: Israeli', 'N NOM SG: air']], ['Action-Name:', 'kill', 'Action-Status:', 'Past Particle', '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']]]
Triplet 2	[[['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']]]
Triplet 3	[[['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:', []]]]

3.2 The Knowledgebase

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, out of context, evokes something positive or something negative [26]. Our system maintains a list of scored verbs, adjectives, adverbs, nouns and named entities. For instance, ‘destroy’ usually bears a negative connotation, whereas ‘develop’ has a positive connotation. Cognitive and commonsense knowledge resources have been utilized to assign prior valence to the lexicon entries, and the resources also leverage scoring of new words, as explained in the following paragraphs.

Scoring a list of Verbs, Adjectives and Adverbs. A group of eight judges have manually counted the number of positive and negative senses of each word of our selected list¹ of verbs, adjectives, and adverbs according to the contextual explanations of each sense found in WordNet 2.1 [2]. A judge’s score of a verb is stored in the format: verb-word [Positive-Sense Count, Negative-Sense Count, Prospective Value, Praiseworthy Value, Prior Valence]. The Prior Valence, Prospective and Praiseworthy values indicate the lexical affinity of a word with respect to “good” or “bad”, “desirable” or “undesirable”, and “praiseworthiness” or “blameworthiness”, respectively. Prospective and Praiseworthy values of the words are not used in this system (we use it for the detailed emotion analysis).

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. E.g., 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 collected the scores for all the listed words.

Formula 1 assigns a prior valence (i.e., a value between -5 to 5) to each selected word. A subset of verbs (e.g. like, love, hate, kiss etc.) from the verb list is marked by a tag named < affect > to indicate these verbs have strong affective connotation regarding preference or dislike. This subset is formed according to WordNet-Affect [22].

$$c(w) = \frac{\sum_{i=1}^m ((\frac{p_i - n_i}{N_i}) * 5.0))}{m} \quad (1)$$

Here, $c(w)$ = Prior Valence of word w , whereby $-5 \leq c(w) \leq +5$

m = Number of judges (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

To measure inter-agreement among judges, we used Fleiss' Kappa statistic [4]. The Kappa value for the prior valence assignment task for 723 verbs, 205 phrasal verbs, 237 adjectives related to shape, time, sound, taste/touch, condition, appearance and 711 adjectives related to emotional affinity and 144 adverbs is reliable ($\kappa=0.914$).

¹ <http://www.englishclub.com/vocabulary/>

Moreover, our scoring resembles to the EVA function [6] 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.

Scoring of Nouns. Since manual scoring is a tedious job and there exist more nouns than the above lists, we employed ConceptNet [10] to assign prior valence to nouns. ConceptNet is a large semantic network of commonsense knowledge which encompasses the spatial, physical, social, temporal, and psychological aspects of everyday life. A value from [-5,5] is assigned as the valence to an input noun or concept. (Here we use noun and concept synonymously.) To assign valence to a concept, the system collects all concepts which are semantically connected to other concepts and from other concepts to the input concept found in the ConceptNet. The returned entries are separated into two groups depending on their semantic relations. The entries of the first group correspond to relationships like ‘IsA’, ‘DefinedAs’, ‘MadeOf’, ‘PartOf’, etc. and the second group entries corresponds to relations like, ‘CapableOf’, ‘UsedFor’, ‘CapableOfReceivingAction’, etc. Of the two groups, the first one basically indicates other associated concepts, and the second one indicates the actions that the input concept can either perform or receive. The first list is searched against the scored list of nouns and the first 5 unique concepts which are found in the target list are taken from the matching list. An average score of those matched 5 concepts is returned as the valence of the non-scored concept. If the first procedure fails to assign a valence, a similar procedure is performed for verbs.

Let us look at an example. In the case of the noun ‘doctor’, the system initially 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.

Possible_concept_list = ['person', 'smart person', 'human', 'conscious being', 'man', 'wiley bandicoot', 'clever person', 'dentist', 'pediatrician', 'surgeon', 'physician', 'veterinarian', 'messy handwriting', 'study medicine', 'job']

Possible_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 ‘Possible_concept_list’, and failed to assign a value. Therefore, the second list, ‘Possible_action_list’, is processed and from that 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 in the database for future use. We scored about 4500 concepts using this procedure. This list is maintained to speed up processing time since otherwise the system has to invoke ConceptNet every time.

Scored-list of Named Entity. The system also maintains a list of scored named entities. The information of an entity is stored as the following format: Named-entity [Role, Concept, Genre, General-Sentiment, Prior-Valence]. The field ‘Role’ indicates any of the values from the list {*Company, Concept, Country, Object, Other, Person, Product, Service, Team*} and ‘Concept’ stores a ConceptNet keyword to represent the concept of the entity. ‘Genre’ indicates any of the 15 genres (e.g. Politics, Sports,

Technology etc.) taken from the Yahoo! news domain. ‘General-Sentiment’ contains either a negative (-1) or positive (+1) value based on the value of the prior valence towards the named entity. We did not use any named entity recognizer to identify a named entity, and make the simplifying assumption that anything for which ConceptNet fails to assign valence is a named entity. To assign General-Sentiment we have developed a tool that can extract sentiment from Opinmind [28]. For example, ConceptNet fails to assign a valence to “George Bush” or “Asimov”. From Opinmind we get -1 (37% positive, 63% negative) and +1 (97% positive, 3% negative) for those two entities, which is stored as: George Bush {Person, President, Politics, -1, (+1.85, -3.15)}; Asimov {Product, Machine, Science, +1, (+4.85, -0.15)}. Initially a list of 2000 entries is manually created and scored using Opinmind. Usually the value of ‘General Sentiment’ is idiosyncratic and arguable. If the valence-sign of the ‘Concept’ and ‘General-Sentiment’ (e.g. President [+3.533], George Bush [-1]) differs from each other, the system considers this as an ambiguity and assigns neutral valence to the sentence referring that named entity.

3.3 Contextual-Valence and Sentiment Assessment

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

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

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

$T_1 = \langle \langle \text{President, Concept, \{the\}}, \text{call, past, \{time: tuesday\}}, \text{discovery, Named-Entity, \{the, space, shuttle\}} \rangle \rangle$

$T_2 = \langle \langle \text{President, Concept, \{the\}}, \text{wish, infinitive, \{dependency\}}, \text{astronaut, Concept, \{the, adv: well\}} \rangle \rangle$

$T_3 = \langle \langle \text{President, Concept, \{the\}}, \text{congratulate, infinitive, \{dependency\}}, \text{astronaut, Concept, \{dependency, goal: space walk\}} \rangle \rangle$

$T_4 = \langle \langle \text{President, Concept, \{the\}}, \text{invite, infinitive, \{dependency\}}, \text{astronaut, Concept, \{dependency, place: white house\}} \rangle \rangle$

Knowledgebase. The knowledge-base of the system has been discussed in Section 3.2. Using that data source, the system builds the following computational data-structure that is consulted to process the input text. The verbs are classified into two groups, affective verb (AV) and non-affective verb (V) group. The verbs having the tag $\langle affect \rangle$ in the knowledge-base are members of AV. Both AV and V are further partitioned into positive (AV_{pos} , V_{pos}) and negative (AV_{neg} , V_{neg}) groups on the basis

of their prior valences. Similarly, adjectives (*ADJ*), adverbs (*ADV*), concepts (*CON*) also have positive and negative groups indicated by ADJ_{pos} , ADJ_{neg} ; ADV_{pos} , ADV_{neg} ; and CON_{pos} , CON_{neg} ; respectively. For a named entity (*NE*) the system creates three kinds of lists, namely ambiguous named entity (NE_{ambi}), positive named entity (NE_{pos}) and negative named entity (NE_{neg}). The named entity that has a different sign for the valence of ‘concept’ and ‘general sentiment’ fields is a member of NE_{ambi} .

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

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

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

Pseudo Code for Processing:

Begin

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

 tripletSet_{*i*} = getSemanticParsing (S_i)

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

 for each triplet T_j in tripletSet_{*i*} do //we assume $1 \leq j \leq m$, m triplets

 actorValence = ContextualValenceAttrib (actorPriorValence, actorAttributes)

 actionValence = ContextualValenceAttrib (actionPriorValence, actionAttributes)

 objectValence = ContextualValenceAttrib (objectPriorValence, objectAttributes)

 actionObjectPairValence = setActionObjectPairVal (actionValence, objectValence)

 tripletValence = setTripletValence (actorValence, actionObjectPairValence)

 tripletValence = handleNegationAndConditionality (tripletValence, T_j)

 tripletDependency = if the token “dependency” is found then ‘true’ else ‘false’

 tripletResult_{*j*} = {tripletValence, tripletDependency}

 loop until all triplets are processed

 contextualValence = processTripletLevelContextualValence (tripletSet_{*i*})

 sentimentScore = average($\sum_{k=1}^m \text{abs}(\text{contextualValence}_k)$)

 valenceSign = get ResultantValenceSign(contextualValence)

 SentenceValence_{*i*} = sentimentScore * valenceSign

loop until all sentences are processed

valence = getParagraphValence (SentenceValence)

outputValence = valence \cup {SentenceValence}

End

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

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

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

- $ADV_{pos} + (AV_{pos} \text{ or } V_{pos}) \rightarrow \text{pos. Valence}$ (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.

- Neg. Action Valence + Pos. Object Valence \rightarrow Neg. Action-Object Pair Valence (e.g., kill innocent people, miss morning lecture, fail the final examination, etc.)
- Neg. Action Valence + Pos. Object Valence \rightarrow Pos. Action-Object Pair Valence (e.g., quit smoking, hang a clock on the wall, hate the corruption, etc.)
- Pos. Action Valence + Pos. Object Valence \rightarrow 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}) that uses 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 2 out of 5. 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 ‘tagged negative triplet valence’ values for the actor (robber) where the ‘action-object pair valence’ for [“arrive, car”] and [“mug, store-keeper”] are positive and negative, respectively. For such cases where a negative valenced actor is associated with at least one ‘negative action-object pair valence’, the tagged output is marked with a highly negative valence.

But if a negative valenced actor is associated with all positively scored ‘action-object pair valence’ the ‘tagged negative triplet valence’ is toggled to positive. For example, “*The kidnapper freed the hostages and 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 valence’. This implies that an action done by a negative-role actor is not necessarily always negative. We also consider the cases of negation and conditionality as discussed in [5,26].

3.4 Sentiment Assessment

In the previous section we described how valence is assigned to a ‘Triplet’. Here we explain how sentiment is assessed for a sentence. The data structure ‘tripletResult’ has a field named ‘tripletDependency’ to indicate inter-dependency between Triplets. If interrelated Triplets are found, the function *setContextualValence* is invoked to set the contextual valence for those Triplets. If there is no dependency then the variable ‘ContextualValence’ is kept the same as the ‘tripletValence’ of that triplet. But if there are two negatively valenced inter-dependent triplets, their valences are averaged and the sign of the ‘tripletValence’ is changed to positive.

For example, the sentence “*It is difficult to take bad picture with this camera.*” produces two dependant triplets [‘it’, ‘is’, ‘difficult’] and [‘camera’, ‘take’ ‘bad-picture’] and both produce negative valences (-10.00 and -11.945) but the final valence is set to positive (10.973). The average of the absolute values of ‘contextualValence’ is assigned as the ‘sentimentScore’ for a sentence, S . The ‘valenceSign’ is set either +1 or -1 according to the sign of the valence whose value is the maximum among ‘contextualValence’. The value of ‘sentimentScore’ is multiplied with ‘valenceSign’ to get ‘sentenceValence’ and it is the valence we get for a sentence. According to the scoring system the range of ‘sentenceValence’ is ± 15 .

4 System Evaluation

We evaluated our system to assess the accuracy of sentence-level sentiment recognition when compared to human-ranked scores (as “gold standard”) for two data-sets. The first one, Data-Set I, was created by collecting 200 sentences from internet based sources for reviews of products, movies, and news (Yahoo! News, 2006), and email correspondences. It was scored by 23 human judges according to positive, negative,

or neutral sentiment affinity by an online survey². The second set, Data-Set II, is the sentence polarity dataset v1.0³ first used by Pang and Lee [14]. A summary of our data sets is given in Table 2.

Table 2. Input Data-Sets

Sentences	Data-Set I	Data-Set II
Number of Positive Sentences (x)	90	5331
Number of Negative Sentences (y)	87	5331
Number of Neutral Sentences (z)	23	0
Fleiss' kappa (κ)	0.782	

For Data-Set I the number of positive, negative, and neutral sentences has been decided according to the average scores assigned by the judges. The agreement ($\kappa=0.782$) among the judges can be seen as reliable.

We have performed two types of comparisons. First we compared system performance with human ratings, and then with a similar, state-of-the-art system [9] for both data sets. Three types of experiments (E1C1; E2C2; E3C3) are conducted considering three conditions where the ranges for a neutral sentiment are respectively $\pm 16.66\%$, $\pm 23.33\%$ and $\pm 30.00\%$. The motivation behind this is to set-up the decision logic to classify neutral sentences. On both data sets, the accuracy measure for positive sentences (P1) is the percentage of the number of positive sentences that both humans and our system identified as positive. Similarly P2 and P3 are obtained for negative and neutral sentences. The system's overall accuracy is computed as the average of P1, P2 and P3. There is no P3 score for Data-Set II (see Table 3). The results of all the experiments obtained an average of 0.654 and 0.673 as Human-System agreement (i.e., the Cohen's kappa score for Data-Set I and Data-Set II).

Table 4 shows the results for the performance comparison between our system and Liu's system [9]. Although Liu's system does not directly assess sentiment of text, it appears to have outstanding performance to analyze emotion from text of smaller input size (e.g. a sentence). We consider Liu's system [9] because like ours, it is a rule based system, and it seems to be the best performing system for sentence-level emotion sensing. On the practical side, it is freely available on the internet, so we could use it 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 Liu's system can recognize, and for each sentence a vector containing the percentage value afferent to each emotion is returned. We took the highest percentage value from the positive and negative emotion group for each input sentence of our data sets. This resulted in an average accuracy of 81.64% for our system, and 71.75% for Liu's system, when compared to the scores of the human judges as "gold standard".

² <http://ita.co.jp/research/survey/> (one can login using a guest username).

³ Introduced in Pang and Lee at ACL 2005. at <http://www.cs.cornell.edu/People/pabo/movie-review-data/>

Table 3. System Accuracy Metrics using Data-Set I and Data-Set II

Run #	Data-Set I					Data-Set II				Row
	P1	P2	P3	Avg.	κ	P1	P2	Avg.	κ	Avg.
E1C1	88.88	86.21	60.86	78.65	0.649	79.47	85.81	82.64	0.653	80.65
E2C2	85.55	79.31	78.26	81.04	0.701	84.93	83.87	84.40	0.689	82.72
E2C3	76.66	70.11	91.30	79.36	0.613	87.07	80.43	83.74	0.677	81.55

Table 4. Accuracy Comparison Metrics [Liu’s system doesn’t apply any condition]

Run #	Data-Set I		Data-Set II	
	Our Sys	Liu’s Sys	Our Sys	Liu’s Sys
E1C1	78.65%	70.83%	82.64%	72.67%
E2C2	81.04%	70.83%	84.40%	72.67%
E2C3	79.36%	70.83%	83.74%	72.67%
Avg.	79.68%	70.83%	83.59%	72.67%

5 Conclusion

The system described in this paper proposes a 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 robust because we have employed both cognitive and commonsense knowledge to assign prior valences to the words and developed the rules. A study demonstrated the accuracy of the system when compared to human performance, but also, that it outperforms a state-of-the-art system (under simplifying assumptions). In future we plan to compare our results with other machine learning approaches.

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 emotion recognition system, which may classify named emotions rather than positive or negative sentiments. 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

1. Esuli, A., Sebastiani, F.: Determining the semantic orientation of terms through gloss analysis. In: Proc. CIKM, pp. 617–624 (2005)
2. Fellbaum, C. (ed.): WordNet: An Electronic Lexical Databases. MIT Press, Cambridge, Massachusetts (1999)
3. Fitriani, S., Rothkrantz, L.J.M.: Constructing Knowledge for Automated Text-Based Emotion Expressions. In: Proc. CompSysTech (2006)

4. Fleiss, J.L.: Measuring nominal scale agreement among many raters. *Psychological Bulletin* 76(5), 378–382 (1971)
5. Hu, M., Liu, B.: Mining and summarizing customer reviews. In: Proc. KDD (2004)
6. Kamps, J., Marx, M.: Words with Attitude. In: Proc. 1st Intl. WordNet Conf. (2002)
7. Kim, S.M., Hovy, E.H.: Identifying and Analyzing Judgment Opinions. In: Proc. HLT-NAACL 2006, ACL, pp. 200–207 (2006)
8. Kim, S.M., Hovy, E.H.: Automatic Detection of Opinion Bearing Words and Sentences. In: Companion Volume to the Proceedings of the 2nd IJCNLP (2005)
9. Liu, H., Lieberman, H., Selker, T.: A Model of Textual Affect Sensing using Real-World Knowledge. In: Proc. IUI, Miami, FL, January 12–15, pp. 125–132. ACM, New York (2003)
10. Liu, H., Singh, P.: ConceptNet: A Practical Commonsense Reasoning Toolkit. In: Liu, H., Singh, P. (eds.) *ConceptNet: A Practical Commonsense Reasoning Toolkit*, BT Technology Journal, vol. 22(4), pp. 211–226. Kluwer Academic Publishers, Dordrecht (2004)
11. Machine Syntax, the official website (2005), <http://www.connexor.com/connexor/>
12. Mihalcea, R., Liu, H.: A corpus-based approach to finding happiness, Computational approaches for analysis of weblogs. In: AAI Spring Symposium (2006)
13. Nasukawa, T., Yi, J.: Sentiment Analysis: Capturing Favorability Using Natural Language Processing. In: Proc. K-CAP, pp. 70–77. ACM Press, New York (2003)
14. Pang, B., Lee, L.: Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In: Proc. ACL, pp. 115–124 (2005)
15. Pennebaker, J.W., Mehl, M.R., Niederhoffer, K.: Psychological aspects of natural language use: Our words, our selves. *Annual Review of Psychology* 54, 547–577 (2003)
16. Polanyi, L., Zaenen, A.: Contextual valence shifters. In: Shanahan, J., Qu, Y., Wiebe, J. (eds.) *Computing Attitude and Affect in Text: Theory and Applications*, The Information Retrieval Series, vol. 20, pp. 1–10 (2004)
17. Riloff, E., Wiebe, J., Wilson, T.: Learning Subjective Nouns Using Extraction Pattern Bootstrapping. In: Proc. CoNLL-2003 (2003)
18. Shaikh, M.A.M., Prendinger, H., Ishizuka, M.: SenseNet: A Linguistic Tool to Visualize Numerical-Valence Based Sentiment of Textual Data. In: Proc. ICON, pp. 147–152 (2007)
19. Subasic, P., Huettner, A.: Affect Analysis of Text Using Fuzzy Semantic Typing. *IEEE Transactions on Fuzzy Systems* 9(4), 483–496 (2001)
20. Stock, O., Strapparava, C.: Getting Serious about the Development of Computational Humor. In: Proc. IJCAI, pp. 59–64 (2003)
21. Turney, P.: Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews. In: Proc. 40th Annual Meeting of the ACL, pp. 417–424 (2002)
22. Valitutti, A., Strapparava, C., Stock, O.: Developing Affective Lexical Resources. *PsychNology Journal* 2(1), 61–83 (2004)
23. Wiebe, J., Mihalcea, R.: Word Sense and Subjectivity. In: Proc. ACL-06, pp. 1065–1072 (2006)
24. Wiebe, J.: Learning subjective adjectives from corpora. In: Proc. AAI (2000)
25. Wiebe, J., Wilson, T., Cardie, C.: Annotating expressions of opinions and emotions in language. *Language Resources and Evaluation* 39(2-3), 165–210 (2005)
26. Wilson, T., Wiebe, J., Hoffmann, P.: Recognizing Contextual Polarity in Phrase-Level Sentiment Analysis. In: Proc. HLT/EMNLP. ACL, pp. 347–354 (2005)
27. Opinmind, <http://www.opinmind.com/>