A Framework for Extensible Emotion Analysis System

Rekha S. Sugandhi, Aneesh S. Mulye, and Vijay M. Wadhai

Abstract—**The analysis of emotions can be performed on various sources of input, namely: text, speech or voice, images, and videos. This paper discusses a graph-based technique that prepares the ground for flexible and expandable emotion analysis from text documents. We take into account the complexities of human emotion representation in text that is more challenging and susceptible to inconsistent outcomes. The design of the proposed system aims towards a generalized solution for such kinds of text mining applications and emphasizes a simplified (yet flexible) emotion representation and analysis system that can be subjected to fine-tuned analysis, depending on future requirements. The aim of our work is to develop a prototype that can be used for (unsupervised) learning from some given text and can be used for possible extension of the emotion dictionary.**

Index Terms—**Affective Computing, Natural Language Processing, Opinion Mining, Pattern Clustering, Text Analysis, Text Mining.**

I. Introduction

In general text mining or text summarization, the major tasks involve finding the predominant senses of the words occurring in a given document. However, opinion mining a document, i.e. the task of identifying opinions, sentiments, or emotions from a given source text can be an extremely challenging one. This is often because direct references to words may not be sufficient to extract sufficient information about human thoughts. Specifically, affects expressed in the input text may not be perceived consistently depending on the filtering techniques used. For example, the sentence, "She lowered her eyes on hearing this." might not be expressive enough to indicate the involved emotion with regard to the statement. It is important to weigh these words with relevance and intensity measures to decide the dominance of the sentiments expressed. The "lowering of eyes" mentioned in the example statement, may be indicative of "shyness", "feeling of shame", "disappointment", or may be "sadness".

There are various sources of input (namely text, speech, facial expressions, body language, etc.) that exhibit human affects. This paper considers text document as the source input to be analyzed to identify the (set of) emotions or sentiments expressed.

Opinion mining applications are often required to analyze a document for affects and summarize them at various levels of the text content, namely, as a topic feature, or at different contexts in the document body (for example, to identify the affects exhibited in only some specific paragraph). At such a time, the analysis model should be tunable on parameters like block-level or document-level analysis, the choice of prevalent 'n' number of affects in the document in a certain order of relevance, the monitoring of variations in the exhibited aspects throughout the flow of the document's content, and so on. Hence, a flexible and robust model that can cater to the above tuning parameters is needed.

In this paper, a model is proposed that takes a text document as its input and analyzes it to give the various emotions that it is related to. The system initially represents the organization of emotions in a graphical-cum-taxonomical form. This necessarily includes the partitioning of all known emotions into sets and subsets that follow the human perceptions of these emotions. It is hence presumed that a certain pre-defined set of thoughts and perceptions shall be provided in the form of an emotion dictionary that will assist the system in analyzing the input document for the related emotions expressed in it. This important assumption is made since the method of categorizing emotions depends on the intended purpose of the analysis. The emotions that are considered relevant in an input source would be different for psycho-analysis, weblog mining for commercial products, analyzing candidates for job interviews, and so on. Apart from the pre-defined emotion dictionary, the system requires a Machine Readable Dictionary (MRD) as the knowledge base that will be required to identify the correct sense in which a certain word in the text is used in. In addition to sense induction, the MRD will also be required to relate to each other the various words occurring in the same document in the same context. The system gathers the various expressive words in the document (context-wise) and generates a semantic inter-relation among these words in the form of a graph with meaningful nodes and links. The values attached to the nodes and links are then evaluated for the emotion analysis. The paper is organized as follows: Section II discusses the related systems in the area of opinion and emotion analysis. Section III provides justification of the graph-based approach that is being implemented for the proposed framework. Section IV explains the architectural framework of the approach. The steps in implementation are explained in Section V.

Manuscript received June 21, 2011, revised September 22, 2011.

Rekha S. Sugandhi is with the MIT College of Engineering, Department of Computer Engineering, Pune, India, (e-mail: rekha.sugandhi@mitcoe.edu.in).

Aneesh S. Mulye was Persistent Systems Ltd., India.He has enrolled for MS(Computer Science) from the Georgia Institute of Technology, Atlanta USA (e-mail: aneesh.mulye@gmail.com).

Dr. Vijay M.Wadhai is with the MIT College of Engineering, Pune, India as Principal, He is also Director, Research and Development, Intelligence Radio Frequency (IRF) Group, Pune. (e-mail: wadhai.vijay@gmail.com).

II. Related Work

Earlier researchers have tried various approaches to solve problems related to affective computing through AI techniques, like support vector machines, Hidden Markov Models, Bayesian networks. and simple taxonomical approaches. Primitive efforts in this direction involved polarity detection on terms involved in the corpus. This was done by considering a basic set of words assigned with positive and negative polarity and then extending the set of words by considering the words whose synsets contained words of known polarity; that is, assigning positive (negative) polarity to new words that contain positive (negative) terms in their synsets.

Further work in emotion recognition also involved categorizing the text document in one of the major categories of emotions, namely *anger, fear, happiness, sorrow* and the like. The method involves training the learning system on a set of labeled text data and the conversion of text words into keywords that are assigned indices (taken from the Vector Space Model) and values proportional to their frequency [2]. Few approaches towards opinion mining are feature-based, where certain topics or features are known before the data sets are trained. Generally, such methods are applied to specific attribute-level opinion mining (like weblogs containing customer opinions/reviews of products). Also, in such cases, learning is implemented at the sentence level, wherein the sentence is split into clauses to check for conditionals or comparative phrases, each of which is given specialized treatment. The same technique does not work for general sentences other than conditional or comparative sentences [7], [8]. These methods are indicative of opinions, not necessarily affects, and hence are more suitable for applications involving the mining of opinions in documents, blogs, etc. rather than general emotion analysis. The methods applied in most earlier work are vector-based, where emotions are represented in a taxonomical structure. The earlier work (Esuli and Sebastini, 2007) describes the method of ranking synsets representing opinions based on pruning parameters as per the random-walk model in the PageRank algorithm. In this method, WordNet synsets are represented by a graph model where the synsets are represented as nodes and a directed link from a synset node n_A to a synset node n_B indicates that the gloss of n_A includes some terms belonging to synset n_B. The random-walk PageRank Algorithm when executed on the graph identifies words ranked on positivity and negativity. Since two runs (one for the positivity ranking and one for the negativity ranking) are done on the data, this method successfully ranks some opinions that may be both positive and negative, as decided by the context. This work also relates more to the polarity ranking of opinions [1].

The structure of the text document can also be related to feature extraction from them. The prevalent emotions can be predicted from their relative location in the document, namely the opinions expressed in the first and last paragraph; this sounds logically reasonable, since paragraphs represent contextual blocks. The various emotions identified in the text are assigned weights depending on their locations and occurrences in the text, after which the effective weights are calculated to identify the emotion expressed in the article [3].

III. The Graph Based Approach

The task of natural language processing for text involves the evaluation of semantics as per the grammar rules and structural constructs of the language. It is hence necessary to represent the understanding of the constructs in a way that semantic links are clearly evident and comprehensible. Using graph-based methods for such representations seems a good choice, since graphs contain nodes and links to which arbitrary quantities of information may be attached.

A. Appropriate Representation for the Problem Domain

We can use the graph-based method to represent the information extracted from the text that needs to be analyzed. In the system, potential words (words that may be emotive) are represented as nodes, and all such potential words in a block are connected with links if they are semantically related. A graphical form of representation makes it simpler to show the analogy between related words. The structural properties of graphs are used here analogously to natural language terms for evaluation and analysis.

A second advantage of using the graph representation is that a separate graph can be generated for each contextual block of text (a paragraph represents a context in the design) due to clusters of word nodes being automatically formed. Each cluster can be treated independently for analytical purposes. The strengths of the related words are indicated by assigning between them appropriate link weights. Two antonymous word nodes are assigned negative link weights with high (negative) values.

B. Classification of Emotions

The list of emotive words – or, rather, words that indicate affects – is very limited. There can however be numerous other words that may not be a direct representation of an emotion but could still be a related word. Examples include verbs or actions that indicate some affects indirectly (e.g. the phrase "*looked elsewhere*" may indicate '*avoidance*'). We organize the set of emotions words with each such word accompanied with its set of synonyms, antonyms, indicative-verbs (actions), and their descriptive gloss. In addition to this, we also assign a value (ranging between -10 and 10) to indicate the valence of the emotions. Here the valence refers to the intensity of the said emotion word and whether it refers to a positive or negative emotion that is directly indicated by the sign of the emotion value assigned. Emotions have been classified based on the general taxonomical structure and notion that there are four pure emotions, namely, *anger, fear, happiness* and, *sadness*. These four pure emotions can be appended to eighteen more general or basic emotions that in our terminology [12] give twenty-two primary emotions [13]. The set of pure and primary emotions thus can be as listed below:

- 4 pure emotions: P_u = {*anger, fear, happiness, sadness*}
- 22 primary emotions: P_r={*aggressiveness, anger, avoidance, compassion, courage, excitement, fear, happiness, helplessness, humility, impatience, indifference, joy, optimism, pain, patience, pessimism, sadness, selfishness, surprise, unselfishness, vigilance* }

The remaining set of emotions can be identified as being part of a set of secondary emotions (P_s) that also includes the twenty two pure and primary emotions. It indirectly indicates pure emotions (P_u) as intense feelings. The secondary emotion set (P_r) refers to a lesser but stronger set of feelings. Further, the secondary set of emotions may represent more specific but (perhaps) less intense feelings.

The idea behind our classification method for the listed (known) emotions is to be able to reach the required level expected from the granularity tuning parameter value. The graph structure now simplifies the relation (linkages) between the various sets of emotions. If new emotions are now to be appended to this existing classification, it becomes a matter of simply understanding the relation (if any) with existing pure or primary emotions. If so, then the new emotion can be added to P_s and then linked to the relevant emotion in its lower layer in the directed acyclic graph (the architecture is explained in section IV).

C. Other Resources

The proposed system requires a knowledge-base for the natural language in which the input text document is expressed. Thus for word sense disambiguation, we need a lexicon. The lexicon that we refer to in our system is the widely-used WordNet. WordNet is used to find the semantic relation in the identified words in order to find their possible links to emotive words or actions. For that matter, the system design allows use of any online dictionary that can be used as a lexical resource.

IV. Architecture

A. Basic Architecture

The graph structure is conceptually built on a single, layered directed acyclic graph (DAG). It grows upwards, with the root(s) at the bottom. Its growth upwards is strictly non-decreasing. An upper layer contains all the nodes from lower ones. A node has a count/weight associated with it; so does a link. The weights on nodes and links are calculated from top to bottom. That is, from one layer to the next (in the downward direction), the new weight value of each node in the lower layer is equal to the old value multiplied by the sum of incoming links times the weight of the higher-level node from which they come. Note that except for those between the top two levels, all downward links are one-to-one; otherwise, the links downward from any node must be normalized to have their weights sum to one. Formally:

Let w_B^L be the weight of the lower node before the summation. Let there be N links incident on this node, with link weights of L_i, i Є [1: N], coming from nodes of weights w_i^H, i Є [1:N] (respectively).

The weight of this lower node after summation is:

$$W_A^L = \sum_{i=1}^{N} (L_i \times w_i^H) \tag{1}$$

The basic idea here is to let the information filter down the graph through the links attached depending on the strength of the links (that is indicated by the link weights). If an important node is linked to a lower-level one only weakly, it has a moderate impact and so on. Usually, all levels except the top have node weights of one, so that all information content is derived from the input, and all emotion information is derived from the pre-assigned link weights.

Strictly speaking, the weight of every node in all layers except the first can be reduced to a weighted linear sum of the weights of the nodes of the first; the above description is intended purely as an aid to visualization, and as a demonstration of how granularity of analysis may be varied.

B. Data Representation

The data involved in the system includes the input text document to be processed for the analysis, the predefined set of emotions in the form of a dictionary, a basic set of emotion graphs (that too is predefined and obtained from the emotion dictionary) and, the content words extracted as result of the pre-processing of the input text. The input text is maintained in its source form. The content words obtained after the pre-processing are maintained as a list of words tagged with their POS-tags. These words are later represented as nodes of the graph. The nodes necessarily are pairs of node names (the lemma) and its occurrence count in the current block of text. Thus, in the design, we have two graph representations. The first one is a static graph that interprets the assumed relations between the various emotion words in the language. The second graph is generated from the word information distilled from the input text. The second (dynamically generated) graph is merged with the more static predefined emotion graph before application of the method of analysis.

C. Extending the Architecture

The emotion graph has been organized as layers of nodes (that resembles an inverted tree data structure) with the height of a layer representing the granularity of emotional classification. There are no links between nodes in the same layer – all links are to the layer beneath. The link weight represents how connected that emotion is to the (more fundamental) one below. The link weights are subjective, and can be modified considering the purpose or motive of the emotion analysis. This dependency is incorporated in the design keeping in mind that sentiment, emotion or opinion analysis have varying purposes like product surveys for sales analysis, psycho-analysis, interviews, to name just a very few.

The emotion graph (G_E) – a multistage directed acyclic graph- has a basic layer of nodes that represent pure or most primitive emotions. These are connected (subjectively, depending on the semantic relations) to the next layer of nodes that include the primary emotions as well as the pure emotions of the basic layer. The next layer is further organized as collection of pure and primary emotion nodes along with the secondary emotion nodes. Strictly speaking, each stage can be derived directly from the first, but the stages are retained for the purpose of displaying the results of analysis at different granularities (Fig. 1).

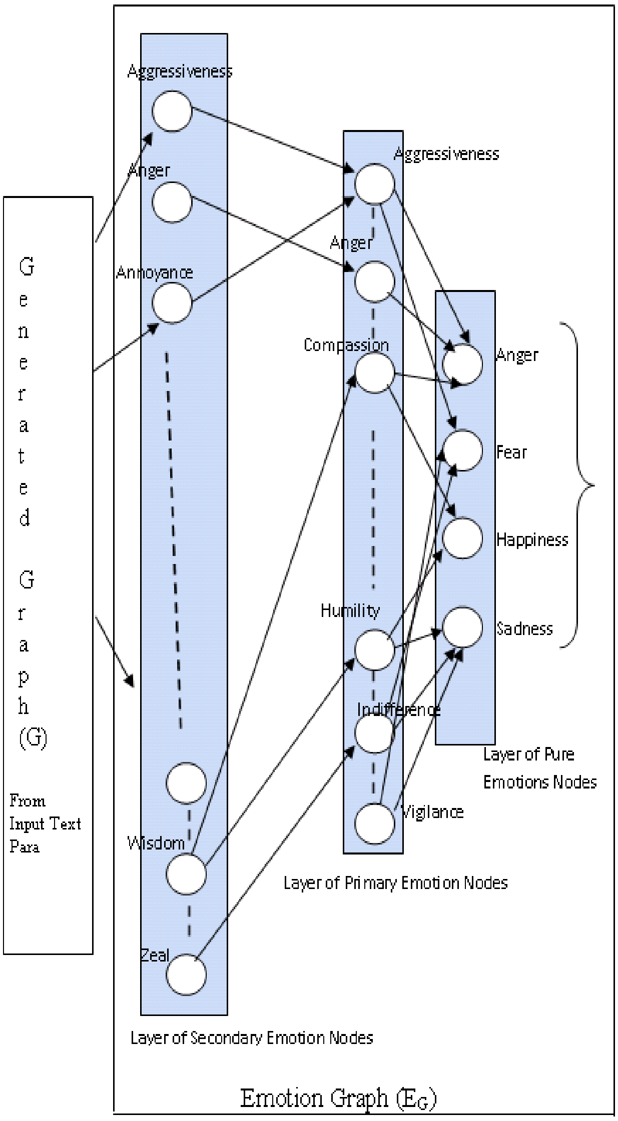

Fig.1. Mapping between generated graph(G) to Emotion Graph

This layered network of nodes has two objectives: ease of construction and modification, and the ability to distill the data in the first layer into later layers. Thus a current layer node can derive values as a function of the nodes of the previous layer it is connected to and its link weight.

As distillation is an operation meaningful only when at least two layers are present, it is layers, not nodes, which contain a reference to the previous layer.

The basic structure (G_E) is extended into a larger graph during the word sense disambiguation and analysis steps. The disambiguated content words from the input document are checked for emotion content and then linked to the predefined set of nodes.

V. Steps for Implementation

A. Basic Steps

The basic steps for the emotion analysis can be (refer to Fig 1):

Step 1: Input text document and split it into paragraphs

Step 2: Split each paragraph into its set of sentences

Step 3: Pre-process each sentence (explained in the next sub-section)

Step 4: Iteratively, convert each pre-processed sentence into graphs (the nodes of which represent potential words/concepts in the input text and the links between the nodes specify the relation between the concepts). *Note:* For every new sentence, the graph is incremented in nodes for newly found words and their corresponding links with other nodes from previous iteration.

Step 5: Perform word sense disambiguation on the nodes [17]. Identify the related senses of the nodes and their weights according to the semantic proximity of the respective nodes with possible emotions.

Step 6: Sort the nodes according to the weights calculated for the nodes and perform the emotion analysis.

B. Pre-processing of Text

Like in all NLP applications, here also it is necessary to pre-process raw input text data and prepare it for further analysis and processing. The input for the pre-processing includes the sentence that is obtained after breaking up paragraphs in the given document .The standard pre-processing tasks [16] that we perform on the data are in the sequence as follows:

1. Tokenization: involves breaking a given sentence into words. The word tokens can now be either individually processed or can be treated as a positional entity in a sentence depending upon the pre-processing task.
2. Part-of-Speech tagging: The Penn tagger is used to attach the part-of-speech to the tokenized words, useful to sense short-listing during the process of sense disambiguation. Penn tags are more specific some of which are not identified in WordNet. Thus after POS-tagging, the named Penn Tags are mapped to one of four basic WordNet tags, namely, noun(n), verb(v), adjective(a), adverb (r) [15].
3. Lemmatization: Involves finding lemmas (base words) for all the words that are tagged.
4. Stop-word removal: Stop-words that do not contribute much towards finding meanings of texts are eliminated. This is done with reference to a pre-defined list of

stop-words taken from WordNet, excluding negation stop-words like nor and not.

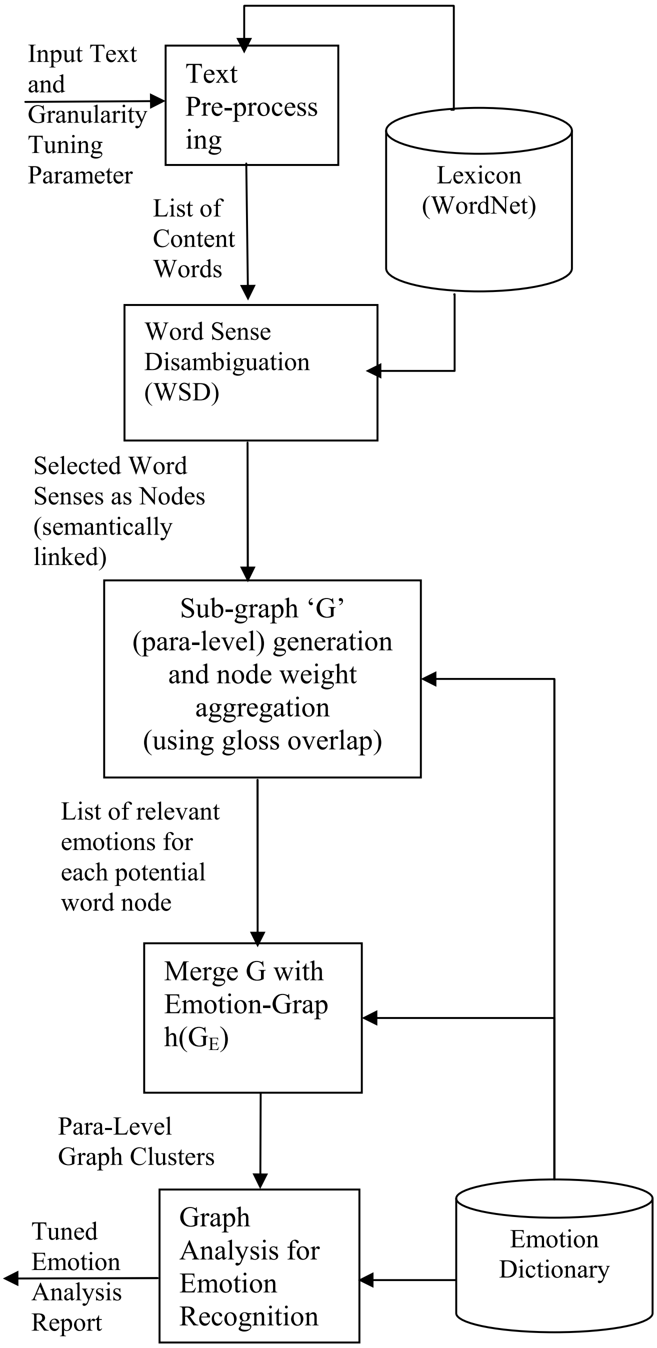
Input Text and Granularity Tuning Parameter
Text Pre-processing
Lexicon (WordNet)
List of Content Words
Word Sense Disambiguation (WSD)
Selected Word Senses as Nodes (semantically linked)
Sub-graph 'G' (para-level) generation and node weight aggregation (using gloss overlap)
List of relevant emotions for each potential word node
Merge G with Emotion-Graph(G_E)
Para-Level Graph Clusters
Graph Analysis for Emotion Recognition
Emotion Dictionary
Tuned Emotion Analysis Report

Fig 2. Proposed Framework for the Analysis

C. Method of analysis

1. Identify context blocks in the input text by breaking the input corpus into paragraph-sized text. This seems logical, since in the text a change in paragraphs indicates some change or variation in the current context.
2. For every such block/paragraph break the paragraph into sentences. For each sentence, do the following:
   i. Perform text pre-processing.
   ii. Extract sense information for all the processed (content) words. Here we use the WordNet senses.
   iii. Build graphs for the identified sense possibilities. Perform sense disambiguation on the generated sense graph [17].
   iv. Append this graph to the other sub-graphs of all sentences in the same block by merging common word nodes or by connecting those nodes of the sub-graphs that are semantically closer. Subsequently this merged graph is connected to the predefined DAG graph that corresponds to the given definitions of emotive words.
   v. Calculate node weights and assign link weights based on the semantic relations between the word nodes connected by a link in the graph.
   vi. Calculate the effective node weights and sort the words based on the weights that the nodes hold.
3. Compare the weights of the emotive word nodes.
4. Based on the granularity required, output the major emotion in the said document, either block-wise or document wise.

The pre-processing of sentences (Step *2(i)*) gives potential words (in their lemma-form) that are called content words. These content words form nodes in the graph that is generated. To connect the content nodes we use the WordNet graph. This includes placing some intermediate word nodes representing common synsets of the two potential nodes. This process of finding common synset nodes is repeated for every two nodes identified for a particular sentence. The identified nodes are connected by links. There is an upper bound L on the maximum path length between two nodes [17]. (L is usually between four and six.)

Step *2(iii)* pertains to finding the most appropriate sense of the extracted word (node) in a particular sentence [17]. For this purpose, we apply certain measures on the nodes. (Note that we are first processing individual nodes at the sentence-level). After each sentence in a block/paragraph is converted into nodes and links, it is attached to the overall graph for the paragraph. The graph is initially empty before a paragraph is processed. One sentence in the current block after conversion to nodes and links is merged with the main graph, before proceeding to the next sentence. The graph keeps growing till all the sentences of the paragraph are converted.

For the current word/concept node, certain *node-based measures* are calculated in order to understand its semantic relevance and affect for emotion recognition. Three node-based measures (to be implemented) are the key-player, node centrality and, node degree.

Key-player measure: The key player (KP) measure aggregates over the minimum path distance over a particular node from all other nodes. Hence this parameter indicates the proximity of one concept node from all others. The node having the highest KP value (its calculation involves reciprocal of the shortest node distance) proves to be a major word in the context/block [18].

$$N_{KP}(u) = \frac{\sum_{v \in V : u \neq v} \frac{1}{d_{uv}}}{|V| - 1} \quad (2)$$

Centrality measure: The centrality measure of a node in an undirected graph indicates the role of a node as a good authority node. A good authority node *N(u)* in turn is a summation value of good hub nodes that are also immediate neighbors of the node *N(u)*. Hence, $N_C(u)$, the centrality measure of a node, *N(u),* is the value obtained from finding mutually recursive values of authority, *a(u)* and hub, *h(u)*, given as

$$a(u) = \sum_{v:(v,u) \in E} h(v) \; and \; h(u) = \sum_{v:(u,v) \in E} a(v) \quad (3)$$

Since, the generated sub-graphs are undirected graphs, the edges in the sub-graph, G, are un-ordered: edges *(u,v)* and *(v,u)* mean the same edges without direction. Hence the *a(u)* and *h(u)* evaluate to the same value. Therefore, the centrality measure for a node u can be totally contributed by either the node authority or the node hub value [18]. Here, we take the node authority measure as the value for the node centrality measure, $N_C(u)$. Therefore,

$$N_C(u) = a(u) = \sum_{v:(v,u) \in E} h(v) \quad (4)$$

Node Degree: The degree of a node ($N_D(u)$) indicates the measure of how well one node is connected to the others. It simply refers to the ratio of the number of outgoing edges from a node to the maximum degree of the graph (|V|).

$$N_D(u) = \frac{|(v,u) \in E : v \in V|}{|V - 1|} \quad (5)$$

For every node *u* in the generated sub-graph, *G*, the node weight is now calculated as a function that takes as parameters, the three node measures namely, key-player value, the centrality measure, and the node degree measure. This, however, is only relevant for nodes in their block context. It is now required to relate these weighted nodes to their corresponding emotions. For this, we apply a simple method to connect the graph nodes with emotive words. We compare the glosses of the word node and the emotive words to find overlaps of potential value [19]. The gloss overlap (*Gl_Overlap*) values are sorted and compared to find the matches with maximal values.

Thus for every node, $u \in G$, the node weight can be calculated as:

$$W(u,E) = F(N_{KP}(u), N_C(u), N_D(u), Gl\_Overlap(u,ED)) \quad (6)$$

The obtained node weight represents relevance measure of word represented by node u with the emotive word list E that is a list sorted in descending order of relevance measure.

D. Merging the Generated Sub-Graph to the Emotion Graph

We use combinations of node measures and gloss overlap measures that are selected to choose the correct sense from the various possible ones for a given target word. Not only those, the calculated measures are also modified for node weight and link weight calculations for further analysis. From the calculated node weights of generated sub-graph G (Eq. 6), we can now extract the words with emotive content, by considering for the node u the relevance measure values listed in *E* above a certain threshold. The shortlisted relevant emotions thus obtained for a particular word node, u, gives a direct mapping from the generated sub-graph to the emotion graph. This can be done in the following simple steps:

- For every node in the sub-graph, obtain the sorted list of related emotions and their relevance measures, *E (from Eq.6).*
- Extract the emotions for the node that satisfy the specified relevance measure threshold. (The threshold value will be randomly determined as function of document length and aggregate on the total number of shortlisted emotion words, E_{sh}, for all nodes in G)
- Connect the node u, of sub-graph G, with each of the short-listed emotions that are nodes in the highest layer of the emotion graph.
- The links between the nodes of G and emotion node in the emotion graph will be assigned weight corresponding to the respective relevance measure from the list, E_{sh}.

We now perform the analysis of the obtained concept graph, paragraph-wise, for paragraph-based emotion distilling, using Eq.1 to obtain relevant emotion words for each paragraph. The first and last context blocks (paragraph) of the document are known to have the most relevant views [2]. Therefore, the sub-graphs pertaining to the first and last paragraphs are emphasized with respect to those of other paragraphs, by assigning additional weights to their nodes in order to elevate their emotional relevance for further distilling. The overall emotion of the entire document can now be treated as an aggregate of those obtained from each paragraph or text block. The approach described here allows one to have optional emotion analysis on whole or sections of the document. In spite of implementing the framework as a graph, we are able to reduce computational overhead because we are implementing its structure as a simple ordered list, with minimal links between the layers in the DAG.

VI. Further Work

In our current proposed framework, we have measured the word-sense relevance using local node properties. We plan to improve the sub-graph analysis by incorporating the overall edge density and graph entropy values to contribute to the effective emotion analysis. Also, relevance measures on word collocations in the context of emotion analysis will be addressed in the subsequent work.

VII. Conclusion

Due to certain inconsistent constructs in the English language, it becomes a challenging task to adapt to the changing contexts in the uses of terms. It becomes even more challenging to process non-technical and abstract terms like opinions and emotions. The framework that we propose in this paper aims to represent an extensible system for specific domains. Dynamic learning techniques can be applied to this framework to help the system evolve for domain vocabulary improvisation.

References

[1] A. Esuli, F. Sebastini, "PageRanking WordNet Synsets: An Application to Opinion Mining", 2007

[2] Z. Teng, F. Ren, S. Kuroiwa , "Emotion Recognition from Text based on the Rough Set Theory and the Support Vector Machines", 2007, IEEE

[3] J. Li, F. Ren, " Emotion Recognition from Blog Articles", 2008 IEEE

[4] J. Yan, D. Bracewell, F. Ren, S. Kuroiwa, "The Creation of a Chinese Emotion Ontology Based on HowNet", Engineering Letters, 16:1, EL_16_1_24, 2008.

[5] A. Abbasi, H. Chen, A. Salem, "Sentiment Analysis in Multiple Languages: Feature Selection for Opinion Classification in Web Forums", ACM Transactions on Information Systems, vol. 26, no. 3, Article 12, June 2008.

[6] Kazuhiro Seki, Kuniaki Uehara, "Adaptive Subject Trigeers for Opnionated Document Retrieval", ACM WSDM'09 Barcelona, Spain, February 9-12, 2009.

[7] M. Ganapathibothla, B. Liu, "Mining Opinions n Comparative Sentences", 22nd International Conference on Computational Linguistics (Coling 2008), Manchester, August 2008, pp. 241-248

[8] R. Narayanan, B. Liu, A. Choudhary, "Sentiment Analysis of Conditional Sentences", In Proceedings of 2009 Conference on Empirical Methods in Natural Language Processing, Singapore, 6-7 Aug 2009, ACL and AFNLP, pp.180-189

[9] W. Du, S. Tan, "Building Domain-Oriented Sentiment Lexicon by Improved Information Bottleneck ", CIKM'09, November 2009, Hong Kong, China, © ACM 978-1-60558-512-3/09/11

[10] R.W.Picard, "Affective Computing: From Laughter to IEEE", IEEE Transactions on Affective Computing, Vol 1, No. 1, January-June 2010, 1949-3045/10/$26.00©2010 IEEE

[11] M. Ptaszynski, P. Dybala, W. Shi, R. Rzepka, K. Araki," Towards Context Aware Emotional Intelligence In Machines; Computing Contextual Appropriateness Of Affective States", In Proceedings of the 21st ACM International Joint Conference on Artificial Intelligence, San Francisco, CA, USA, 2009

[12] R.Sugandhi, R. Bedi, Dr. V.M..Wadhai, "Graph-based Clustering Approach for Emotion Recognition in Text Document", Mineware-2011, National Conference on Data Warehousing and Data Mining, January 2011, pp 60-67

[13] Classification for emotions: http://www.surfyoursoul.com/dictionary.html

[14] B. Liu, "Sentiment Analysis and Subjectivity", Handbook of Natural Language Processing, Second Edition, (Editors: N. Indurkhya and F.J. Damerau), 2009

[15] WordNet- http://wordnet.princeton.edu

[16] Natural Language Toolkit- www.nltk.org

[17] R. Navigli, M. Lapata, "An Experimental Study of Graph Connectivity for Unsupervised Word Sense Disambiguation", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 32, no. 4, pp. 678-692, 2010.

[18] S.P.Borgatti, "Identifying Sets of Key Players in a Network", In Proceedings of Conference for Integration of Knowledge Intensive Multi-Agent Systems, Boston, MA, USA, 1-3 Oct 2003, pp. 127-131.

[19] R. Navigli, "Word Sense Disambiguation: A Survey", ACM Computing Surveys, vol. 41, no. 2, Article 10, 2009.

Rekha S. Sugandhi has completed her graduation (B.E. Computers) from K.K. Wagh College of Engineering, Nashik in 1998 and post-graduation (M.Tech Computers) from Government College of Engineering, Pune in 2006, both under the University of Pune. Her research areas include natural language processing and linguistics, affective computing, machine learning and, data mining. She is currently working as Assistant Professor at M.I.T. college ef Engineering, Pune, India. She is a member of ISTE and CSI.

Aneesh S. Mulye completed his undergraduate degree (B.E. in Computer Engineering) from the MIT College of Engineering, Pune, under the University of Pune, and is pursuing his graduate studies (enrolled for an MS in Computer Science) from the Georgia Institute of Technology, Atlanta. His areas of interest include natural language processing, symbolic computing and non-Von Neumann architectures, and evolutionary theory and genetic algorithms.

Dr. Vijay M. Wadhai received his B.E from Nagpur University in 1986, M.E. from Gulbarga University in 1995 and Ph.D. from Amravati University in 2007. He has an experience of 25 years that includes both academics (10 years) and research (14.7 years). He has been working as a Principal, MITCOE, Pune (since 21st Feb 2011) and simultaneously handling the post of Director, Research and Development, Intelligence Radio Frequency (IRF) Group, Pune (from 2009). He has published 60 papers in various conference and Journals (30 in International Journal, 21 in International conference and 9 in national conferences). He has to his credit two patents in the field of m-health and data mining. His research interest includes Data Mining, Natural Language processing, Cognitive Radio and Wireless Network, Spectrum Management, Wireless Sensor Network, VANET, Body Area Network, ASIC Design, VLSI. He is a member of ISTE, IETE, IEEE, IES and GISFI (Member Convergence Group), India.