Psyche Mining with PsycheTagger – A Computational Linguistics Approach to Text Mining

Ahsan Nabi Khan, Liaquat Majeed Sheikh, Summaira Sarfraz

***Abstract*—** The human elements of personality working behind the creation of a write-up play an important part in determining the final dominant mood of a text. This article is a detailed description of a formal research in Text Mining using purpose-built Computational Intelligence tools, PsycheMap and PsycheTagger. PsycheMap is created to classify documents based on emotive content, while PsycheTagger, is the first semantic emotive statistical tagger in English Language. Working in the lines of statistical Parts-of-Speech Taggers, this tool is adapted to perform efficiently and accurately for emotive content. The tagger self-ranks its choices with a probabilistic score, calculated using Viterbi algorithm run on a Hidden Markov Model of the psyche categories. The results of the classification and tagging exercise are critically evaluated on the Likert scale. These results strongly justify the validity and determine high accuracy of tagging using the probabilistic parser. Moreover, the six-step mining implementation provides a linguistic approach to model semi-structured semantic dataset for classification and labeling of any set of meaningful conceptual classes in English Language Corpus.

***Keywords*—** Emotive content, Semantic Tagger, Computational Linguistics, Text Mining, Classification, Statistical Taggers, Likert scale, Bayesian Techniques

I. Introduction

TEXT Mining is the field related to Natural Language Processing, Information Retrieval and Data Mining in which unstructured data in the form of text is preprocessed and analyzed to find the quality, relevance, novelty and interestingness. Most of the data in organizations is stored in the form of text. The volume is estimated 80-85% of the total data generally but minimal estimates put unstructured data as 35% and semi-structured data as 22% which leaves the structured data in DBMS only 47%. The unstructured data in the form of weblogs, web pages, documents, instant messaging and emails, and the semi-structured data in the form of ontologies and XML files has been predicted to increase rapidly in near future [1]. This increases the commercial importance of text mining, especially of semantic content hidden inside the unstructured and semi-structured text.

Humans communicate with explicit, implicit and subliminal patterns of emotions and of variations in moods. Such patterns play a vital role in effective expression, interaction and interpretation of under-tones and over-tones. Representations of emotive content in the unstructured and semi-structured data have gained recent attention in written and oral communication. The simplest and most widely used one among these is the categorical representation. Categories are based on:

- Evolutionary basic emotion [2]
- Everyday frequent emotion patterns [3]
- Application specific emotions [4]
- Moods and other affective states [5]

HUMAINE Emotion Annotation and Representation Language (EARL) [9] has recently suggested XML realization of emotive content. Since it admits that standardization of a model based on emotive content is non-existent [6], EARL develops dialects of XML schema based on emotion categories, dimensions [7] and appraisals [8] from different sources. Our tagged categorization with probabilities acting as dimensional intensities can be used to fit well into such schemas. Hence, hidden emotive content in unstructured textual data can be used in classification, prediction and other types of analysis and training models and eventually transformed into more meaningful structured content.

Semantic Web of emotive content is also a relevant outcome of related research. Web Ontology Language and Semantic Web Rule Language [28], though sparingly used in current industry and academia, have been working on same lines to organize meaningful content and relationships. Any semantic data can be further normalized and used as a semantic web using SWRL Rule Layer over the top of OWL Layer [29].

The scope of such emotive content is in the mood and sentiment analysis of consumer reviews of products, stock

This paper is the journal follow-up of the Conference Proceedings in The 11th International Conference of Artificial Intelligence, Knowledge Engineering and Databases, Cambridge, U.K. February 22-24, 2012 (AIKED'12).

Ahsan Nabi Khan has graduated magna cum laude from National University of Computer and Emerging Sciences, Lahore and is currently pursuing studies at University of Engineering and Technology, Lahore, Pakistan. He has contributed in five international publications in the field of Computational Linguistics and Data Mining (phone: 92-333-4466846; fax: 888-274-7010; e-mail: ahsan.nabi@gmail.com).

Liaquat Majeed Sheikh, is currently Associate Professor in the Computer Science Department, National University of Computer and Emerging Sciences, Lahore, Pakistan. He has research publications in the field of Data Mining, Algorithms and Emerging Sciences (e-mail: liaquat.majeed@nu.edu.pk).

Summaira Sarfraz is currently managing Humanities Department, National University of Computer and Emerging Sciences and pursuing her PhD studies related to English Language (e-mail: sumaira.sarfraz@nu.edu.pk).

market trends and market basket analysis. Marketing and PR departments would be interested in extracting emotive knowledge of the potential clients or customers. On the macro level, political analysts, economists and the media are highly interested in the variations of moods as coinciding with some specific event or region.

In the Education Sector, the expressiveness of emotion as a technique of pathos, with a touch of ethos and logos, is better learned by students using tools developed to monitor and scale their expressive abilities. Kuo et. al. [27] have demonstrated the success of Big 6 techniques in information problem solving using student and teacher responses on Likert Scale. E-Learning using electronic books is enabled using tools developed for supportive learning.

Our research is the seminal work in the field of social features extraction from human writings. Writers are not secluded from the society; rather they get influenced generally by the following:

- Society: community, family, gender, religion and economics
- Origination: nation, country, city, village, tribe, language, dialect
- Cultural Influence: emotional development and drivers, reward, psychology
- Communication Style: individual, group, business, social, media

What we are covering are the emotional drivers in the third point above in this study. This will link up with other cultural influences to discover the hidden knowledge in the social patterns.

II. Literature Review

Subjectivity at the document level has been studied by various approaches. Lexicon based methods have been used earlier to categorize texts based on subjectivity [11] [12-13]. Various supervised [14-15] and unsupervised [16] data-driven methods have recently been adopted for classification of subjects. Probabilistic models have also been proposed for measuring polarity levels [17]. We see recently that Neural Networks are out-performed by Bayesian Networks by a comparison of 12 percent versus 77 percent accuracy levels in classifying emails using textual features as in Spam and Non-Spam emails [30]. Naïve Bayesian Instance Based Weighting Techniques have been shown better performance over k-Nearest Neighbors algorithm for classifying nominal and numeric datasets [31]. However, such techniques work at structured datasets and not in unstructured expanse of textual content.

Aggregate level subjectivity has also been studied by some. Aggregate features describing customers' praises or complaints in online product reviews are computed by Liu, Hu and Cheng. They also compared opinions of different reviewers [18]. Global mood levels have been captured by one application shown in http://www.moodviews.com . Gilad and Maarten [10] visualize global phenomenon of mood variation with respect to time as reflected by the bloggers. Moods are a highly variable quantity with respect to time. For a blogger, temporal mood may be sad and may change to depression after some interval. Similarly, happiness may transform into relaxed state.

The most common attribute for classifying text is a list of N-grams that most likely indicated particular moods. Corpus is annotated with tags on groups of words that reflected a mood from the author. This enables to identify words and phrases most indicative of the moods by quantifying divergence between term frequencies across various corpora.

One-versus-all (OVA) Support Vector Machines, Regression and Metric Labeling using Markov Random Fields are the three popular techniques to draw a line to separate categorical data. Studies show OVA perform best among these in a complex multi-class data on sentiments [19]. However, more than five classes have not been tested in the study.

The volume of corpus in mood classification should be sufficiently large for high performance (800 training sets produce 48% performance level while 80000 training sets increase performance to 59.67%) [20].

All the cited references work at document level. There has been little study done on sentence and phrasal level mood classification. One such study labels emoticons on extracted emotive content in text [21]. The study points at different sources to look for emotive feature set: including feeling or emotion words; words carrying emotional content e.g. The Company vs. we; textual techniques like pauses, commas, exclamation mark, ellipsis, font size, weight and color; and XML presentations.

Lists of feeling or human emotions are available from many standard and non-standard sources. Robert Plutchik [22] created two lists of eight basic and eight advanced emotions. The basic ones together with their exact opposites are:

Joy vs. Sadness, Trust vs. Disgust, Fear vs. Anger, Surprise vs. Anticipation, Sadness vs. Joy, Disgust vs. Trust, Anger vs. Fear, Anticipation vs. Surprise.

HUMAINE EARL [9] classifies an emotion set of 48 items. Parrot [23] finds a tree structure spreading out of six basic emotional nodes: Love, Joy, Surprise, Anger, Sadness and Fear.

This paper is closely linked to the publication by Sarfraz, S. [25] in which the same list of human emotions is used to label 50 student response essays directed to use emotive content. The methodology used in the publication is a weighted scoring based on frequency of occurrences of psyche words and their synonyms. It has 18 accurate and 20 close-to-accurate document labels that form about 76% accurate results.Please submit your manuscript electronically for review as e-mail attachments.

III. METHODOLOGY

For implementing the PsycheTagger, we needed to go through the following steps:

1. Select a set of psyche tags from emotive words. We selected a comprehensive list of human emotions from the open course contents of Purdue University. [26] It contained 81 human emotions, some of which rarely occurred. Many had overlapping meanings as their synonyms list overlapped. Still they had different connotations attached that their synonyms could not decipher. The list contained all the basic emotions, as given in appendix.
2. Select a corpus containing text in English language. The corpus should contain something comparable to 70 million words according to the standard set in [20]. (In actual, we selected a relatively small corpus of 3 million words from Online Library of Project Gutenberg [24] to keep training time minimal. Novels and scriptures were selected owing to their heavy use of emotive content.)
3. Tag each open-class word with a psyche category. (We used an automated tagging technique based on keywords and synonym matching.)
4. Create transition matrix recording probabilities of going from one psyche state to another. Calculate transition probabilities P(Tag_i|Tag_{i-1}) from the tagged corpus and update in matrix of Tag (row) by Tag (column). (We are using bigrams for the model, so only one tag history is maintained.)
5. Create emission matrix recording observation probabilities of instantiation of a word given a psyche state. Calculate emission probabilities P(Tag|Word) from tagged corpus and update in a matrix of Tag(row) by Word (column).
6. Hidden Markov Model is defined by transition and emission probabilities. Now use Viterbi algorithm to calculate most likely tag sequence given an untagged document.

The following example shows how our Hidden Markov Model output is calculated once it is run over a four-word English input.

The Viterbi algorithm runs from the first word to the last in the input and for each WordX it calculates the product of transition and emission probabilities and multiplies it to the viterbi product for each cell in the previous time order (for Word[X-1]) to update Viterbi[WordX,psycheY]. Once all words are passed, the matrix of viterbi products has the viterbi path – the most probable path of tag sequence as shown in Table 1.

Word1	Glad ( ) = 0.4	Excited ( ) = 0.6	
Word2	Energetic (Excited) = 0.5	Enthused (Glad) = 0.5	Powerful (Excited) = 0.5
Word3	Capable (Enthused) = 0.6	Confident (Powerful) = 0.4	Enthused (Enthused) = 0.4
Word4	Charmed (Capable) = 0.7	Cautious (Confident) = 0.3	

Table 1. Demonstration of internal working of Viterbi Algorithm for psyche tagging

The back-pointers show the most probable sequence of psyches with overall probability 0.7. The psyches in the brackets show the previous word psyche from which it derives the sequence.

The High-Level Architecture Diagram of the Model to implement the methodology is given in Figure1.

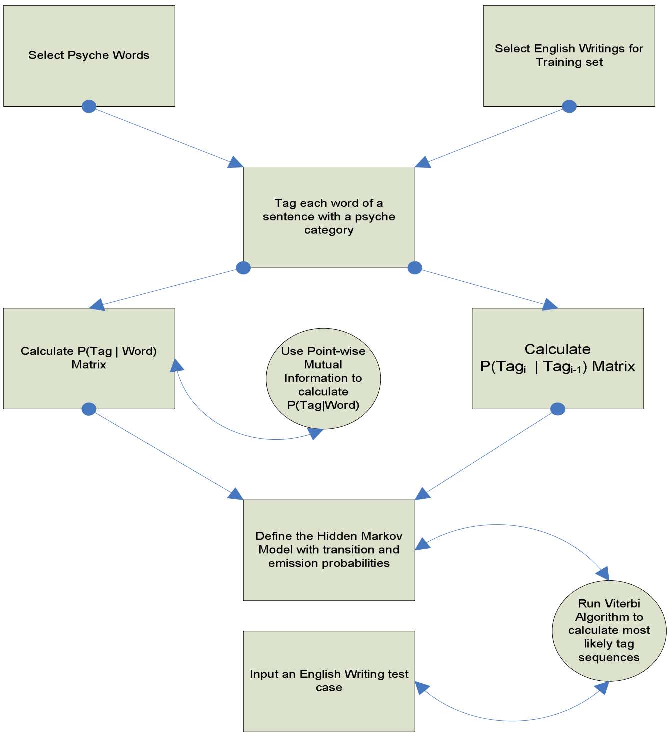
Select Psyche Words
Select English Writings for Training set
Tag each word of a sentence with a psyche category
Calculate P(Tag | Word) Matrix
Use Point-wise Mutual Information to calculate P(Tag|Word)
Calculate P(Tag$_i$ | Tag$_{i-1}$) Matrix
Define the Hidden Markov Model with transition and emission probabilities
Run Viterbi Algorithm to calculate most likely tag sequences
Input an English Writing test case

Fig. 1. High Level Architecture Diagram of Psyche Mining with PsycheTagger http://psychetagger.somee.com

IV. IMPLEMENTATION

Our implementation of the PsycheTagger was a by-product of three-year long study and research in Psyche Mining, an official research project of National University of Computer and Emerging Sciences. (More projects can be seen at http://www.nu.edu.pk). Initially the concept of a psyche-tagger was non-existent but we had the purpose of extracting information from unstructured semantic content in the domain of sentiment analysis. We see that Bayesian Techniques are remarkably superior to Neural Networks and other models in classifying Spam from Non-Spam emails [30]. After several test tools, we created an integrated tool PsycheTagger that used the capabilities of the rest of the components, as detailed in this section.

Different tools that we implemented in the Version 2 of our Psyche Mining Project are outlined as under:

A. PsycheMap

This tool helps to tag any English text document by most likely psyches or moods. It creates dataset of documents against occurrences of psyche words which later may be used to train a model and do association and classification analyses. It uses a formula to calculate total score of a document or paragraph's semantic (psyche) content

Functional Specifications: Identify and record tokens of the English writing text. The token contains information of the lexeme (instance of the word), the category where it belongs, the weight and the count. Construct a queue of the tokens that contain predefined psyche words, synonyms and word variants.

1) makeMap() reads from the predefined psyche words file and constructs the queue
2) mapFile() reads the input file that is any English writing. It will tokenize and count occurrences of psyche words and save the new count in the queue.
3) calcWeight() will calculate the score of overall psyche of documents based on the formula:

Score = W1 x C1 + W2 x C2 + W3 x C3 + … + Wn x Cn,

where W is the predefined weight and C is the count of the word form in the whole document.

Input File "psychewords.txt": This is the file having the list of psyche words, synonyms and variations of word forms and associated weights as defined by linguist and developer. It was saved as a tab-delimited file, to be opened as a spreadsheet.

1) "input.txt": This file contained any English writing text to be tagged.

Output Files

1) "psycheWeightOutput.txt": This file contained the list of psyche words categories (legend) and their respective total score from the document as calculated by the function calcWeight().
2) "checkedWords.txt": This file contained the list of all psyche words synonyms and word forms, associated with the legend, the count and the weight. This file is used to help the tester manually calculate the scores from the respective weights and counts.
3) "dataClip.txt": This file created a dataset for analysis in Data Mining tools such as WEKA. It creates a comma-separated file containing numbers representing scores in the order of psyche word categories as occurred in "psychewords.txt".

B. matrixCreator

matrixCreator is a second version of 'PsycheMap' tool which is used to statistically train the model using huge corpus. It calculates the probabilities of 'psyche-neighboring' words that co-occur with psyche. It also calculates the prior probabilities of psyche occurrences. These probabilities later help in decision making of psyche in next component of model 'PsycheTag' and the Hidden Markov Model 'HMM'.

We have implemented 'matrixCreator' tool for our Psyche Mining Project Version 2 that would be second version of 'PsycheMap' tool. 'matrixCreator' tool is being used to statistically train the model using huge corpus. The training is in a table of lexicon words that fall a specified number of words before the psyche word, called 'psyche-neighboring' words. These words play a huge part in determining the context of the psyche. If they co-occur with the psyche, their probability of co-occurrence is calculated as P(Word | Tag) for each Tag = Psyche[i] , and is saved in the table. These probabilities help us further in decision-making of the psyche in the next component of the model 'PsycheTag'.

Another matrix of prior psyches P(Psyche[i] | Psyche[i-1]) is also calculated from the matrixCreator module. This matrix and the lexicon calculating P(Word | Psyche[i]) aid in calculating the most likely tag sequence in HMM module, using the Bayes Rule.

matrixCreator Background: As an intermediate prototype to be used for 'PsycheTag' module, we try to calculate P(Word | Tag) for each psyche-neighboring word and its psyche tag. It is calculated by

P(Word | Tag) = P(Tag & Word) / P(Tag)
= Count(Tag NEAR Word) / Count(Tag)

As our study of past research shows, the '&' used in the first formula has been interpreted as NEAR operator in searching. NEAR is taken as a distance of 10 words, or any other, as specified as the sliding window parameter.

The probability value ranges from 0 to 1, being closer to 1 for more psyche-biased words, and close to 0 for less emotional content.

Functional Specifications: We simply populate a huge table of Count(Tag NEAR Word) and another table for Count(Tag[i] after Tag[i-1]) while reading the English text file. The first table consists of single words found in the neighboring of psyche words. These are listed in the column. Psyches are listed in rows. These psyches are the legend from a large list of psyche synonyms stored in "psychewords.txt". The second table stores last occurring psyche tag in rows and current occurring psyche tag in the column and increment for each pair.

Here we see that words that co-occur highly with the psyche have high probability. Note that co-occurrence with a psyche word does not mean that we would only be considering the psyche words in the column of the tables above. They are only 78 in number. We however would count the co-occurrence even if any synonym of the 78 psyches occurs with the lexicon words. For that we have a file of over thousand synonyms of psyche words saved in "psychewords.txt". Hence the co-occurrence would not just be in words but in semantic content.

C. PsycheTag

This tool is developed as the second module of the second version of 'PsycheMap' tool for tagging psyches to sentences and would go into finer grains of sentences-level detail than document-level. It uses probabilistic training dataset generated from 'matrixCreator' tool and would test English text for association-mining rules.

It is a more sophisticated tool than first version because it uses probabilistic training dataset "lexicon.txt" generated from the previous component, 'MatrixCreator', to train and then test any English writing text for association rules mining. Specifically, it creates an ARFF file that when run on WEKA generates rules for document-wise psyche dependencies. 'SentenceTag', the third component in this series, takes as input the dataset created by this component tool for another type of sentence-wise rule mining.

'PsycheTag' is a C++ program that deals with high-level document-wise determination of human psyche dependencies. Based on the trained lexicon with probabilities for pre-defined dominant psyches, the tool is to be used for pure probabilistic psyche mining as the second milestone in the implementation phase of our Psyche Mining Project Version2.0.

Background: Taking any word going through an English text, we do not immediately know the likelihood of its usage in the context of a psyche. It is only when the psyche appears in any of its word forms or synonyms that we directly determine the overall mood of the context. Only then we realize that other seemingly neutral words actually are needed to convey the mood. Our test data has shown one sentence:

"No bags, No 7am to 2 pm time limits, no restrictions, no checks and with may other thoughts of freedom i woke up and with all my thoughts focused on this new chapter of my life."

This sentence does not immediately convey moods. However, when the 'PsycheTag' is run on this sentence has determined the following psyches:

Anxious, Bewildered, Concerned, Confident, Depressed, Eager, Excited, Free, Friendly, Good, Great, Impatient, Jumpy, Manipulated, Miserable, Powerful, Respected, Scared, Uneasy, Used. (In the test we took likelihood probability threshold to be above 10 %.)

How do we find so many psyches from such a seemingly neutral word? The answer lies in the likelihood probabilities for lexicon we found in the previous tool 'MatrixCreator'. The process of finding psyches is in three steps:

1) Identifying all the psyche words, that is words that are highly likely to occur in the context of a psyche.
2) Determining the psyche of the phrase including seemingly neutral words
3) Accumulating the psyches to determining the overall psyche of a sentence or a document.

Thus when deciding upon a sentence psyche, we choose a threshold probability above which we accept the psyche as the determinant of the sentence.

This tool is also used to create a dataset of the sentences against the occurrences of psyches. Such a dataset may be used to train a model for association analysis. It can be developed using other tools such as WEKA.

Functional Specifications: The 'PsycheTag' program contains a single simple FileHandler class that contains variables for storing the lexicon probabilities, lexicon words and psyche words. It reads from 'lexicon.txt' the probabilities and based on them it chooses the most likely psyche tags. Then it outputs the psyche tags as 'T' or 'F' on the 'PsycheTagDataset.arff'.

Following are the main functions

1) readFile() reads the input file "input.txt" and after encountering each end of a sentence (full stop), it prints into the "PsycheTagDataset.arff" a row representing the distribution of psyches in the sentence.
2) printBoolRow() is the utility function that prints a row of comma-separated 'T' and 'F' characters representing the psyche distribution in a sentence. This distribution for each sentence is saved in outBool[] array. It prints to "PsycheTagDataset.arff".
3) updateBoolRow() updates the list of psyche distribution whenever each word is encountered. It first checks if the word is in the file "lexicon.txt" and if found, it checks the corresponding position of the psyche in outBool[] array to be 'T'. outBool[] is printed at the end of each sentence.
4) loadLexicon() will load the "lexicon.txt" file at the start of the program. Loading the file means the model is trained with likelihood probabilities.
5) printChecked() prints the same file as "lexicon.txt" in order to confirm that the "lexicon.txt" file loaded by loadLexicon() function is correctly loaded.
6) setLexiconDimensions() determines how many psyches form the columns and how many lexicon words form the rows in by reading the file "lexicon.txt".
7) printARFFHeader() is the code written to print the Header of the ARFF file. It has hardcoded values of psyches which need be changed in case the psyche set is changed.

Input Files

1) "input.txt": This file contained any English writing text to be tagged.
2) "lexicon.txt": This file is generated by the previous component 'MatrixCreator' and is used to train the model with lexicon probabilities. File format is tab-delimited and opens in MS Excel. First row columns show psyches and rows show probabilities for lexicon of words. Lexicon words are listed in first column.

Output Files

1) "psycheTagDataset.arff": It is the dataset in the ARFF format. Each row in the dataset represents the occurrence of psyches in a particular sentence. 'T' means the psyche exists and 'F' means it does not. The same order for psyche distribution is taken as found in "lexicon.txt", that is alphabetical order for 78 psyches. Header is hard-coded, not system-generated.
2) "checkedLex.txt": This file has the same contents and format of "lexicon.txt". It is used to confirm that the file "lexicon.txt" is correctly loaded.

The modules of PsycheTagger were translated from the High-Level Architecture design and methodology into the implementation phase. An ASP.NET Web Application was created and deployed online. It is in public access on the website address http://psychetagger.somee.com . No database was attached to the Program logic. The Application data was loaded from text files into the Page object once the page was loaded.

The simple Graphical User Interface provided a textbox and one button to input and submit test data. For retesting, the page was to be loaded again. The Page load almost always took less than three seconds on Dual Core machines with 3 Mbps connection. The resulting tagged text took longer time to show based on the length of the input text and the number of words in lexicon stored in Emission.words .

Point-wise Mutual Information was used to test the validity of emission probabilities. Once the data proved consistent, the training was directly given from corpus to the matrices using MatrixCreator module. The training produced the emission and transition files that were fed into Transition and Emission objects that defined the Hidden Markov Model in the ViterbiTagger object.

The Graphical User Interface used _Default class instance to communicate, execute and get results from ViterbiTagger object. As the application is run, a new object of ViterbiTagger is created for each _Default page object's PageLoad event that is not PostBack. The user enters the text and submits by pressing the button below the textbox. This triggers the function RunViterbiTagger() which calls the tagText function() of the ViterbiTagger object and returns the tagged string to be displayed in the readOnly textbox in the PostBack event on the Page. The tagText function implements viterbi algorithm. It calls viterbiPass() to find the maximum viterbi value for each open-class word in the sequence. Finally tagText() retrieves the maximum path sequence using backpointers and a stack and returns the string in a user-friendly format of tagged text.

The classes of SynonymList and WeightedPsyche define some additional characteristics of synonyms of mood words and their weights. These characteristics are intended to be used in future work.

The cost of Viterbi algorithm is O(n^3) and internal binary search cost for emission probability is log(m) which gives the final computation cost of O(n^3)log(m) . This can be reduced by partitioning the search-space and pruning the lowest-probability viterbi paths.

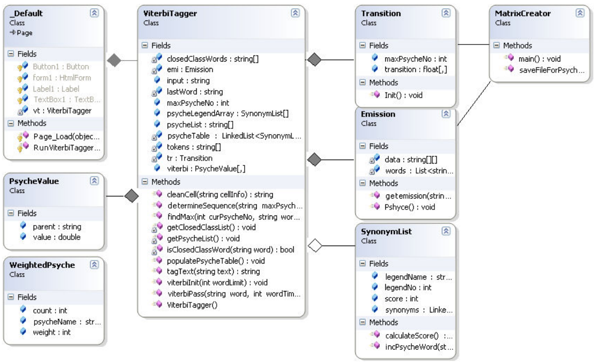

Fig. 2. Class Diagram of PsycheTagger Web Application http://psychetagger.somee.com

V. RESULTS

Ten essays were chosen from the dataset used in the earlier PsycheMap experiment mentioned by Sarfraz in her publication [25]. The essays were divided into two groups. First five of the essays were part of the control experiment that were correctly tagged by PsycheMap, while the other five formed the PsycheTagger performance evaluator group that had earlier been failed to be tagged accurately in PsycheMap tool. The results for PsycheTagger were evaluated using Likert Scale as shown in Table 2.

Highly Disagree	Disagree	Do not know	Agree	Highly Agree
1	2	3	4	5

Table 2. Likert scale used to rate the performance of PsycheTagger

Some sample sentences tagged are:

Sentence 1: I feel now that I am better organized, and more ready to own up to my deeds

Tags for Sentence 1:

feel/(fascinated,0.0148148)
better/(thankful,4.35517701876141E-05)
organized/(good,1.58367733996995E-07)
ready/(manipulated,5.72062001309273E-10)
deeds/(eager,1)

Sentence 2: But I am sure as the years go by this will also be clarified and soon I will be getting a positive feedback.

Tags for Sentence 2:

sure/(concerned,0.0411457931187749)
years/(confident,0.000188152423720238)
clarified/(great,1.04818584990822E-12)
soon/(wary,2.97104641888455E-15)
getting/(good,3.20640171737715E-17)
positive/(great,1.11166873489905E-19)
feedback/(confident,1)

The rating is taken from three independent evaluators, two linguists and one student. Standard deviation of the ratings for any particular topic sentence averaged 0.5367 showing a high degree of agreement in the independent rating, even though the evaluators did not share or discuss the evaluation.

The Likert scale rating given by the evaluators is mostly 4-5 as shown by the lightest shade in the contour map. The average rating for all essays is 4.22 out of 5 which shows 84.4% accuracy in PsycheTagger results.

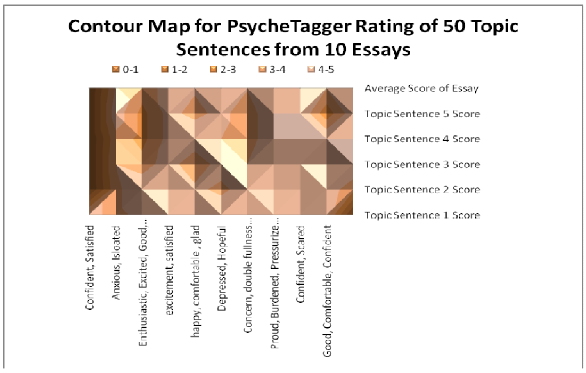

Fig. 3. Likert Scale rating results of fifty topic sentences as represented by fifty square regions. Each row represents the sequence of topic sentence in the essay, 1 being the earliest. Each column represents the essay on the mood expressed as given in the horizontal axis.

The essays used in the corpus have the composition as shown in the example in Table 3.

Essay Expressed Mood	Topic Sentence number	Topic Sentence
Anxious, Isloated	2	I feel now that I am better organized, and more ready to own up to my deeds
Concern, double fullness (not tagged)	5	But I am sure as the years go by this will also be clarified and soon I will be getting a positive feedback.
Proud, Burdened, Pressurized (not tagged)	4	Right from the first day up till now, I have found that the teachers of FAST are very cooperative and strict regarding the rules and regulations of the University.
Excitement, Satisfied	3	Today, after 4 years, I look back and I feel like I paved my own way to where I am, in which I feel great.
Anxious, Isolated	3	University policy helps a little to promoting a healthy amount of socialization between the students.
Confident, Scared	5	The University is now a major part of my life.

Table 3. Sample Essay characteristics of the corpus

For the shown composition of essays and topic sentences, the tags in the resultset are shown in Table 4. The quality of the tags is evaluated by independent evaluators who gave the rating on the Likert Scale from 1 to 5.

Topic Sentence	Tagged Output of PsycheTagger	Average Evaluation
I feel now that I am better organized, and more ready to own up to my deeds	feel/(fascinated,0.0148148) better/(thankful,4.35517701876141E-05) organized/(good,1.58367733996995E-07) ready/(manipulated,5.72062001309273E-10) deeds/(eager,1)	5
But I am sure as the years go by this will also be clarified and soon I will be getting a positive feedback.	sure/(concerned,0.0411457931187749) years/(confident,0.000188152423720238) clarified/(great,1.04818584990822E-12) soon/(wary,2.97104641888455E-15) getting/(good,3.20640171737715E-17) positive/(great,1.11166873489905E-19) feedback/(confident,1)	5
Right from the first day up till now, I have found that the teachers of FAST are very cooperative and strict regarding the rules and regulations of the University.	right/(fascinated,0.00856165) day/(thankful,1.75394638367086E-05) till/(good,1.42813025748599E-07) teachers/(great,3.18681192354752E-10) fast/(good,1.39983692474344E-12) cooperative/(great,2.33091040192122E-20) strict/(guilty,8.87811728995414E-23) regarding/(wary,4.22766584272149E-25) rules/(thankful,1.69476319881573E-27) regulations/(good,2.61932926273239E-29) university/(great,1)	5
Today, after 4 years, I look back and I feel like I paved my own way to where I am, in which I feel great.	today/(free,0.0156028270398274) 4/(friendly,1.56146847846085E-10) years/(frustrated,6.88655360048883E-13) look/(great,7.02493358501951E-16) back/(good,4.56617680503546E-18) feel/(great,1.68961559900911E-20) paved/(guilty,1.24810178755895E-22) feel/(great,4.61833244627151E-25) great/(guilty,1)	3
University policy helps a little to promoting a healthy amount of socialization between the students.	university/(carefree,0.0221674588845372) policy/(good,0.000134064094794705) helps/(great,1.73305322590868E-06) little/(friendly,5.44042777623974E-09) promoting/(great,2.26455479086214E-11) healthy/(miserable,7.96011462142802E-14) amount/(good,1.18633443755557E-15) socialization/(great,6.90110918152337E-24) students/(great,1)	3.33
The University is now a major part of my life.	university/(fascinated,0.0357143) major/(thankful,0.000334062102668475) life/(guilty,1)	2.33

Table 4. Evaluation of PsycheTagger output against expressed moods in essays and their topic sentences

VI. Discussion

The results of PsycheTagger show remarkable improvement to those of the study by Sarfraz, S. using PsycheMap tool [25]. PsycheTagger shows consistent results for both the groups of essays: the control group and the PsycheTagger performance group.

The first row of Table 4 shows the essay which had good rating for both PsycheMap and PsycheTagger. The second and third essays were not tagged by PsycheMap because the human labelers did not label correctly. Here, PsycheTagger tagged better than even the human labelers and scored high rating from the linguists. The fourth and the sixth essays were tagged correctly as great and thankful until the last incorrect tag of guilty. "Guilty" has some probabilities artificially inflated and can be toned down to a more natural level by techniques like smoothing for such rare categories. The fifth essay shows the limitation of our model which cannot detect understatement with words like "little".

As the contour map suggests, topic sentences are more likely to represent the subject and mood if found earlier in the essay. The first essay with the expressed mood "Confident, Satisfied" did not rank much due to high rating error (Variance was 1.5 for Topic Sentence 4 rated as 2, 5 and 4). The reason is that it was the first essay to be ranked, the raters were beginning to form their opinion and rating was not normalized to keep it transparent. The essays with sharp difference in emotions like "Confident, Scared", "Excitement, Satisfied" and "Depressed, Hopeful" were rated best albeit minor inaccuracies. Reason was that these essays used comparison and contrast very well and were focused on the immediate emotion rather than an artificial aura of prevalent mood over the whole essay.

The self-ranking mechanism often worked consistent with the rating. As Table 4 results show the high probability at a word position compared to all probabilities of psyche tags at the same word position is for those tags which are highly rated by the human raters. For example, the fourth tags of the five topic sentences in the first essay showed degradation of rating together with that of probabilities, as seen below:

calm/(confident,1.15868E-09) → average rating = 4.33
floor/(guilty,1.72237E-09) → average rating = 4
talking/(relaxed,2.73787E-15) → average rating = 3.67
increased/(wary,6.90565E-16) → average rating = 3.67
quizzes/(wary,9.66704E-27) → average rating = 4.33

This means PsycheTagger can be transformed to self-improve its tagging by giving the user the option of K-best tags within a probability threshold and learn to associate the tag chosen to the observed word. Moreover, some context-based and domain-specific rules can be added to convert this purely statistical tagger into a transformational-based tagger.

VII. Conclusion

We conclude from the results and discussion of PsycheTagger performance that it is entirely feasible to implement semantic psyche tagging using Viterbi algorithm for reaching above 80% accuracy that is close to the golden standard of human labelers. Any other feature set may be used, like Parrot's [23] or Plutchik's [22], to tag customized psyche categories, provided adequate and accurate corpus training is given. Some rules concerning negation and adjective-adverb combinations can be added to alter calculation of viterbi path probabilities to cater for understatements, overstatements and modification in the primary semantic content.

Appendix

Table 3 lists the final psyche categories found after association and classification analysis and used in PsycheTagger as the tagset.

Able	Dumbfounded	Isolated
Adequate	Eager	Jealous
Agonized	Energetic	Jumpy
Annoyed	Enthused	Mad
Anxious	Exasperated	Manipulated
Apprehensive	Excited	Miserable
Bewildered	Exhausted	Obnoxious
Bold	Exhilarated	Overwhelmed
Bored	Expectant	Peaceful
Brave	Fascinated	Pleasant
Burdened	Free	Powerful
Calm	Frustrated	Pressured
Capable	Glad	Proud
Cautious	Good	Relaxed
Charmed	Great	Relieved
Cheerful	Guilty	Sad
Comfortable	Happy	Satisfied
Competitive	Harassed	Scared
Concerned	Helpful	Shocked
Confident	Hesitant	Suspicious
Confused	Hopeful	Tired
Depressed	Hostile	Uncomfortable
Destructive	Ignored	Uneasy
Determined	Impatient	Used
Disgusted	Indifferent	Wary
Distracted	Inspired	Weary
Doubtful	Intimidated	Wasteful

Table 5. List of human emotions (taken from Purdue University Online Courses [26])

Acknowledgment

This research paper is the drawn from the findings and work of Final Year Project at FAST-National University of Computer and Emerging Sciences, thanks to the other project members, namely Ishrat Aleemjee, Daniyal Tarique Naqi, Muhammad Asif and Muhammad Adil who helped in the implementation and literature review. Also thanks to Ali Zulfiqar, the English Language lab instructor, who joined Sumaira Sarfraz and Ahsan Nabi Khan in rating the results. Finally, thanks to Purdue University for providing the Humans Emotions List online as an open list.

References

[1] Russom, P., BI Search and Text Analytics. TDWI Best Practices Report, Second Quarter 2007.

[2] Ekman, P. (1999). Basic emotions. In Tim Dalgleish and Mick J. Power (Ed.), Handbook of Cognition & Emotion (pp. 301–320). New York: John Wiley.

[3] Douglas-Cowie, E., L. Devillers, J-C. Martin, R. Cowie, S. Savvidou, S. Abrilian, and C. Cox (2005). Multimodal Databases of Everyday Emotion: Facing up to Complexity. In Proc. InterSpeech, Lisbon, September 2005.

[4] Steidl, S., Levit, M., Batliner, A., Nöth, E., & Niemann, H. (2005). "Of all things the measure is man" - automatic classification of emotions and inter-labeler consistency. ICASSP 2005, International Conference on Acoustics, Speech, and Signal Processing, March 19-23, 2005, Philadelphia, U.S.A., Proceedings (pp. 317--320).

[5] Scherer, K.R. (2000). Psychological models of emotion. In J. C. Borod (Ed.), The Neuropsychology of Emotion (pp. 137–162). New York: Oxford University Press.

[6] Scherer, K. et al., 2005. Proposal for exemplars and work towards them: Theory of emotions. HUMAINE deliverable D3e, http://emotion-research.net/deliverables

[7] Cowie, R., Douglas-Cowie, E., Savvidou, S., McMahon, E., Sawey, M., & Schröder, M. (2000). 'FEELTRACE': An instrument for recording perceived emotion in real time, *ISCA Workshop on Speech and Emotion, Northern Ireland* , p. 19-24.

[8] Ellsworth, P.C., & Scherer, K. (2003). Appraisal processes in emotion. In Davidson R.J. et al. (Ed.), *Handbook of Affective Sciences* (pp. 572-595). Oxford New York: Oxford University Press.

[9] http://emotion-research.net/projects/humaine/earl/

[10] Gilad Mishne and Maarten de Rijke, Capturing Global Mood Levels using Blog Posts. In: *AAAI 2006 Spring Symposium on Computational Approaches to Analysing Weblogs (AAAI-CAAW 2006)*, March 2006. Also presented at the *16th Meeting of Computational Linguistics in the Netherlands* (CLIN 2005).

[11] V. Hatzivassiloglou & J. Wiebe. Effects of adjective orientation and gradability on sentence subjectivity. In Proceedings COLING 2000, 2000

[12] J. Kamps, M. Marx, R. Mokken, & M. de Rijke. Using WordNet to measure semantic orientations of adjectives. In Proceedings LREC 2004, volume IV, pages 1115–1118, 2004.

[13] S. Das & M. Chen. Yahoo! for Amazon: Extracting market sentiment from stock message boards. In Proceedings APFA 2001, 2001

[14] K. Dave, S. Lawrence, & D. Pennock. Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. In Proceedings WWW 2003, 2003.

[15] B. Pang, L. Lee, & S. Vaithyanathan. Thumbs up? Sentiment Classification Using Machine Learning Techniques. In Proceedings EMNLP 2002, 2002.

[16] P. Turney. Thumbs up or thumbs down? semantic orientation applied to unsupervised classification of reviews. In Proceedings ACL 2002, 2002.

[17] K. Nigam & M. Hurst. Towards a robust metric of opinion. In The AAAI Symposium on Exploring Attitude and Affect in Text (AAAI-EAAT), 2004

[18] B. Liu, M. Hu, & J. Cheng. Opinion observer: Analyzing and comparing opinions on the web. In Proceedings WWW 2005, pages 342–351, 2005

[19] Bo Pang, Lillian Lee. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. 2005 ACL

[20] Gilad Mishne. Experiments with Mood Classification in Blog Posts. In: *Style2005 – the 1st Workshop on Stylistic Analysis of Text for Information Access*, at *SIGIR 2005*, August 2005.

[21] Walt Froloff. System and Method for Embedment of Emotive Content in Modern Text Processing, Publishing and Communication. 2006. Patent US 7,089,504 B1

[22] Plutchik, R. "The Nature of Emotions". American Scientist. July-August, 2001.

[23] Parrott, W. (2001), Emotions in Social Psychology, Psychology Press, Philadelphia.

[24] http://www.gutenberg.org

[25] Accuracy of Text Psyche Based on Moods Interpretation, Sarfraz, S., VDM Verlag Dr. Müller, 2010, ISBN 978-3-639-23547-0

[26] www.tech.purdue.edu/Ols/courses/ols388/collins/human_emotions.doc

[27] Kuo, L.; Yang, S.; Hu, W.; Wu, C.; Yang, H.; Lin, H. *Applying Big 6 on Digital Book for Supporting Learning*, In Proceedings The 9th WSEAS International Conference on ARTIFICIAL INTELLIGENCE, KNOWLEDGE ENGINEERING and DATA BASES (AIKED'10), Cambridge, UK. February 20-22, 2010. pp 303-308

[28] Horrocks, I.; Patel-Schneider, P. F.; Boley, H.; Tabet, S.; Grosof, B.; Dean, M. *SWRL: A Semantic Web Rule Language Combining OWL and RuleM*, W3C Member Submission, URL http://www.w3.org/Submission/SWRL/, 2004

[29] Ahmedi, L.; Jajaga, E. *Normalization of Relations and Ontologies.* In Proceedings The 9th WSEAS International Conference on ARTIFICIAL INTELLIGENCE, KNOWLEDGE ENGINEERING and DATA BASES (AIKED'10), Cambridge, UK. February 20-22, 2010. pp 419-424

[30] Mohamad, M.; Salleh, K. A. *Independent Feature Selection as Spam-Filtering Technique: An Evaluation of Neural Network.* In Proceedings of the 10th WSEAS International Conference on Computational Intelligence, Man-Machine Systems and Cybernetics (CIMMACS '11). Also in Proceedings of the 10th WSEAS International Conference on Information Security and Privacy (ISP'11) . Jakarta, Indonesia. December 1-3, 2011. pp. 38-47

[31] El Hindi, K.; Abu Shawar, B. *Bayesian-Based Instance Weighting Techniques for Instance-Based Learners.* In Proceedings The 11th WSEAS International Conference on ARTIFICIAL INTELLIGENCE, KNOWLEDGE ENGINEERING and DATA BASES (AIKED'12), Cambridge, UK. February 22-24. pp 243-249