International Journal of Scientific Research in Computer Science, Engineering and Information Technology

© 2017 IJSRCSEIT | Volume 2 | Issue 4 | ISSN : 2456-3307

"Matsar" Sentiment Analysis

Yashodhara V. Haribhakta*, Sunil Barhate

Department of Computer Engineering and IT, College of Engineering, Pune, Maharashtra, India

ABSTRACT

Matsar is Marathi language word and the meaning is *envy, jealousy, negativity* and *distrust*. The following sentence *"He got this project because of luck rather than his work"* conveys *Matsar* sentiment. In this competitive world, it increases negativity and jealousy among people due to the success of other people or status of other people etc. Generally it seems between friends, co-workers, businesses etc. It will be helpful in schools, colleges, organizations to get to know the relation between two individuals. In this work, we have proposed a pattern based approach to detect whether the text contains *Matsar* Sentiment or not. The proposed approach gives accuracy of 99% using KNN algorithm, 98% using DT algorithm, 97% using SVM algorithm and 94% using NB algorithm.

Keywords: *Matsar*, Sentiment Analysis, Patterns, Pos-Tagging

I. INTRODUCTION

Sentiment Analysis is a process of extracting information from the text. It is also known as opinion mining. The aim of Sentiment Analysis is to detect the sentiment, opinion, attitude from the text. Lot of research has been done in analyzing sentiments in the text, such as, subjectivity detection, polarity detection, aspect based sentiment summarization, sarcasm detection, spam detection, emotion detection, mood detection, attitude detection, wish detection, sentiment summarization, etc.

Subjectivity Detection implies whether the text contains any sentiment or not, that is, whether text sentiment is subjective or objective. It is unlike of detecting polarity of text. Some remarkable research works on subjectivity detection are [1] [2]. In Sentiment Prediction, sentiments are detected from the text, i.e. positive, negative or neutral. Sentiment detection is done at document level, sentence level and phrase level. Some remarkable research works of Sentiment Prediction are [3][4]. Aspect Based Sentiment Summarization is bit ahead of sentiment analysis. The summarizer creates opinion summary about feature or aspect of products. For example, a Phone has different features like screen, sound, design, etc., then the objective is to give an outline as rating scores on each of these features. Some remarkable research works on this domain are [5] [6] [7]. Text Summarization for Opinion helps to create summary format for text summaries. Some remarkable research works on this domain are [8] [9]. Some reviews may be more helpful compared to others. Instead of displaying user reviews in sequential order, sorting reviews by its helpfulness would improve user productivity. Predicting Helpfulness aims at automatically predicting the helpfulness of user reviews instead of just relying on users ratting. Some remarkable research works on this domain are [10] [11].

In this competitive world, it is observed that, usually, individuals do not appreciate success of others. They carry jealous feelings for their subordinate success. The text *"Ram got this project because of luck rather than his hard work"* is a *Matsar* text, the speaker is not giving importance to Ram's hard work and trying to project that Ram got the project because of luck and not due to his hard work.

The aim of the proposed work is to detect *Matsar* sentiment in English language text. The approach used in this work is extraction of pattern based features defined as *"Ordered sequence of content words(CW) and General Expression Words(GEW)"*. The CW are *Matsar* sentiment contextual words whereas GEW can be replaced by general expressions. This work focuses on pattern creation, determining maximum length of pattern and their effects, finding optimal value of α parameter which is used to optimize the score of partial matching, and finding optimal value of maximum length of pattern to detect *Matsar* sentiment in text.
Since no dataset is available for *Matsar* sentiment, manually the data has been generated for detecting *Matsar* and *non-matsar* text. So a dataset comprising of

320 *Matsar* and 320 *non-matsar* instances has been created.

II. RELATED WORK

While most researchers focus on assigning sentiments to documents, others focus on more specific tasks such as finding sentiments of words, subjective expressions, subjective sentences, and topics. To automate Sentiment analysis, different approaches have been applied to predict the sentiments of words, expressions or documents.

May All Your Wishes Come True: This work deals with study of wishes and how to recognize them using pattern based approach[14]. It detects whether there is a WISH in the product reviews or not. There are two WISH detectors.

a) Manual: Simple phrases like *"i wish ..", "i hope .." etc.* are sufficient to detect WISH in text.
b) Words: In this method they have designed word based text classifier. Using labeled training WISH dataset to train SVM. Finally, they have proposed Automatic Discovery WISH Templates technique. The main idea of this approach is to reduce redundancy in WISH dataset which contains same WISH phrases. The bipartite graph is formed from the WISH dataset. The nodes of bipartite graph are content nodes and template nodes. Using *Template* and *Template+Word* as feature the SVM is trained. The result is 73% using Template as feature and 80% using *Words+Template* as features. Used dataset is customer product reviews and political discussions. Generated domain independent templates can improve the performance of WISH detection. It is very difficult to get annotated training dataset.

Wishful Thinking Finding suggestions and buy wishes from product reviews: [15] Proposed a method which automatically detects the WISH from text documents(english) like customer reviews. They introduced two types of WISH, these are **a) Suggestion Wishes** and **b) Purchasing Wishes**. The proposed approach is based on the rule based method. It identify two types of Wishes. a) Customers desire to see in the improvement in products and b) Buying wish. The input to the system is WISH dataset, POSLEX and NEGLEX. WISH sentences contains phrases which contains modal verbs like "would", "could", "should", etc. So, the rules formed are **as a) modal verb + auxiliary verb + positive opinion word. b) modal verb + preference verb etc.** To find buying WISH the rule phrase contains *"want to ..", "desire to .." etc.* Used dataset is customer product reviews and political discussions. It is first work to find the buying wish of customers. [14] falsely gives wish to the sentences which does not contains any wish, eg. *"I shouldn't have been cheap, should have bought a Toshiba".* This sentence does not contains any desire. This false acceptance is overcome in this approach. Some general wishes which does not fit in "product suggestion wishes" even though it fits in rules.

Pattern based Approach for Mining Users Opinion from Kannada Web Documents: [16] Work have proposed a new pattern based approach to detect the sentiment in Kannada language reviews of products. In this approach they have created a patterns of two words window based upon the pos tagging of text. They have considered the opinion related pos like, JJ(adjectives), NN(noun), CD(cardinal), RB(adverb) and VBP/VBN/VBD(different forms of verb) in the customer reviews. This approach successfully gives the correct opinion of text containing negators. Finally they have showed that this new pattern based approach is much better than the Turney patterns for Kannada language dataset. Used dataset is customer reviews in English and Kannada languages. Performance is better than existing Turneys approach.

A Pattern-Based Approach for Sarcasm Detection on Twitter: [17] have proposed pattern based technique to detect sarcasm on twitter. Different types of sarcasm are defined with four sets of features. The sarcastic tweet dataset is collected by twitter API which includes #sarcasm. In this approach they have classified the tweet words into two categories i.e., CI which contains words which are content important and GFI which contains words which are grammatical function important. The words in tweet which belongs to CI category are lemmatized and the words which belong to the GFI category are replaced with specific expression. Patterns are extracted from this new generated text using minimum and maximum length of patterns. These patterns are used as features to train SVM, KNN and Maximum Entropy algorithms. The achieved accuracy is 83.1% which is much better than the previous approaches. Twitter dataset is used in this work. In [12] work, needed already build user knowledge base, but in this approach doesn't need it.

Performance is better than the previous approaches. This pattern based approach does not have covered all types of sarcastic patterns.

Above research work motivated to build a pattern based model to detect *Matsar* Sentiment using natural language processing.

III. DESIGN MODEL

There are five steps in this approach namely: Data Gathering, Preprocessing, Pattern Creation, Training Modules and Classification.

A. Data Gathering:

A very few times people expresses *Matsar* Sentiment on twitter, facebook, microblogs, emails etc. It is very hard to collect this kind of text on social media. We have tried to collect these kind of text on social media but, we haven't got useful text. So, manually built dataset for *Matsar* Sentiment. We have manually built 320 *Matsar* text. For examples: *"He has many patents than me but he has only in one field where as i have in many fields", "I lost the election because of faulty EVMs". etc.*

B. Data Preprocessing:

In dataset text, it contains various punctuation marks. For our approach we are not considering the context getting from punctuation marks. So, punctuation marks are useless in this approach.

C. Pattern Creation:

Designed algorithm for pattern extraction from text is shown bellow

Algorithm 1 Pattern Extraction Algorithm

```
procedure PATTERNEXTRACTION(text)
    $minL \leftarrow minimum\_pattern\_length$
    $maxL \leftarrow maximum\_pattern\_length$
    $PatternVector \leftarrow []$
    $PatternSet \leftarrow \{\}$
    $cleanText \leftarrow RemovePunctMarks(text)$
    while $word$ in $cleanText$ do

        if posTag($word$) in $CW\_class$ then
            $PatternVector$.append(stem($word$))
        else
            $temp \leftarrow posTag(word)$
            $PatternVector$.append(GeneralExpressionWord[$temp$])
    for $i \leftarrow minL$ to $maxL$ do
        $t \leftarrow$ ngram($PatternVector$, $i$)
        $PatternSet$.add($t$)
    return $PatternSet$
```

In *Matsar* dataset, words are divided into two categories using PoS tags. First category words are Content Words (CW) which are important to get *Matsar* context and second category words are General Expression Words (GEW). The aim of these two category words is to make patterns less specific. The ordered sequence of these two category words gives the *Matsar* Semantic. These words are divided into these two categories using Pos-tag of words. The motivation of dividing words using Pos-tag into two categories is [13]. How to divide words into two categories using Pos-tag is given in TABLE-I.

TABLE I
PART-OF-SPEECH TAG CLASSES.

POS Tag	Description	Class
CC	coordinating conjuction	CW
CD	cardinal number	GEW
DT	determiner	CW
EX	extential there	CW
FW	foreign word	GEW
IN	prep./sub. conjunction	CW
JJ	adjective	CW
JJR	adjective, comparative	CW
JJS	adjective, superlative	CW
LS	list maker	GEW
MD	model	GEW
NN	noun, singular or mass	GEW
NNS	noun plural	GEW
NNP	proper noun, singular	GEW
NNPS	proper noun, plural	GEW
PDT	predetermined	CW
POS	possessive ending	CW
PRP	personal pronoun	GEW
PRP$	possessive pronoun	GEW
RB	adverb	CW
RBR	adverb, comparative	CW
RBS	adverb, superlative	CW
RP	particle	CW
SYM	Symbol	GEW
TO	to	CW
UH	interjection	GEW
VB	verb, base form	CW
VBD	verb, past tense	CW
VBG	verb, gerund/present partiCWple	CW
VBN	verb, past partiCWple	CW
VBP	verb, sing. present, non-3d	CW
VBZ	verb, 3rd person sing. present	CW
WDT	wh-determiner	GEW
WP	wh-pronoun	GEW
WP$	possessive wh-pronoun	GEW
WRB	wh-adverb	GEW

For example: *"He got this project because of luck rather than hard work".*

After Pos Tagging: *He_PRP got_VBD this_DT project_NN because_IN of_IN luck_NN rather_RB than_IN hard_JJ work_NN ._.*

After Pos-tag, words are divided into two categories i.e. CW and GEW. Stemming is done on CW and the GEW are replaced by certain expressions which are given in

TABLE-II.
After stemming CW and replacing GEW with expressions,
Pattern Vector*: [PRONOUN get this NOUN because of NOUN rather than hard NOUN]*
We define a pattern as an ordered sequence of Content words and General Expressions. The patterns of different lengths are extracted from the training set.

TABLE II
EXPRESSIONS USED TO REPLACE THE WORDS OF GEW.

PoS-tag	**Expression**
“CD”	[CARDINAL]
“FW”	[FOREIGNWORD]
“UH”	[INTERJECTION]
“LS”	[LISTMARKER]
“NN”,“NNS”,“NNP”,“NNPS”	[NOUN]
“PRP”,“PRP$”	[PRONOUN]
“MD”	[MODAL]
“RB”,“RBR”,“RBS”	[ADVERB]
“WDT”,“WP”,“WP$”,“WRB”	[WHDETERMINER]
“SYM”	[SYMBOL]

The range of pattern lengths i.e. the minimum (L_{Min}) and maximum (L_{Max}) is given as:

$$L_{Min} <= Length(pattern) <= L_{Max}$$

The number of pattern lengths are $N_L = (L_{Max} - L_{Min} + 1)$. Therefore, from the example mentioned above, we can extract the following patterns:
- *[PRONOUN get this NOUN]*
- *[get this NOUN because of NOUN]*
- *[this NOUN because of NOUN rather than hard NOUN]*
- *etc*.

The patterns which occurs at least N_{occ} are only stored. We have taken $N_{occ} = 1$. In addition to this, duplicate patterns are avoided and also the patterns which appears in non-*Matsar* dataset are avoided. The patterns are stored in sets according to the lengths of patterns. For example all patterns of length three are stored in Pattern Set of length three and so on.

D. Pattern Features

To calculate Pattern Features we have used match degree function match(p_{ij} , t). Where, p_{ij} is pattern *i* from pattern set *j* and *t* is pattern vector of text.

$$match(p_{ij}, t) = \begin{cases} \textbf{1}, \textit{ if pattern appears exactly in pattern} \\ \textit{vector of text.} \\ \alpha * \textbf{n/N}, \textit{ n out of N words of pattern} \\ \textit{appears in pattern vector in same order.} \\ \textbf{0}, \textit{ if no word of pattern appears in} \\ \textit{pattern vector.} \end{cases}$$

where, α is parameter to optimize partial matching score. The value of the features F_{ij} shown in TABLE-III and calculated as:

$$F_{ij} = \beta_j * \sum_{k=1}^{K} match(p_k,\ t)$$

where, β_j is weight given to pattern of length *j* and is calculated as,

where, *j* is length of pattern.

TABLE III

$$\beta_j = \frac{j-1}{j+1}$$

1	F_{11}	F_{12}	...	F_{1N}
2	F_{21}	F_{22}	...	F_{2N}
...	...	...	...	...
M	F_{M1}	F_{M2}	...	F_{MN}

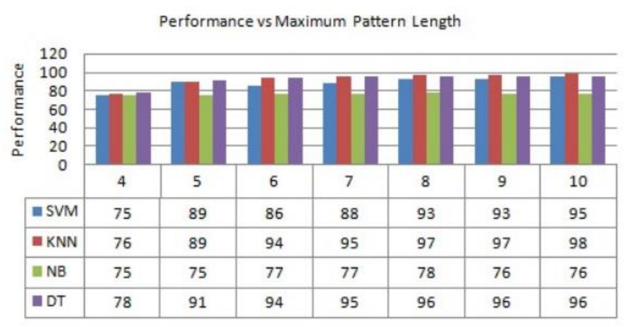
IV. EXPERIMENTAL RESULTS

The keys used for evaluating performance are as follows:
Accuracy: It is ratio of correctly classified tuples to the total tuples.
Precision: It implies that the number of tuples that have successfully classified as *Matsar* over the total number of tuples classified as *Matsar*.
Recall: It implies that the number of tuples that have successfully classified as *Matsar* over the total number of *Matsar* tuples.

A. Effect of Maximum Length of Pattern

For all experiment we kept minimum pattern length as 3 because there is minimum length of text in dataset is 3 and all the experiments are performed as five-fold cross-validation.

All those patterns with maximum length matching with pattern vector are considered as important patterns. So, find the maximum pattern length which gives better performance. For this the α value is kept to one constant for all maximum pattern lengths and the performances of algorithms with different maximum pattern lengths is shown in Fig. 1.



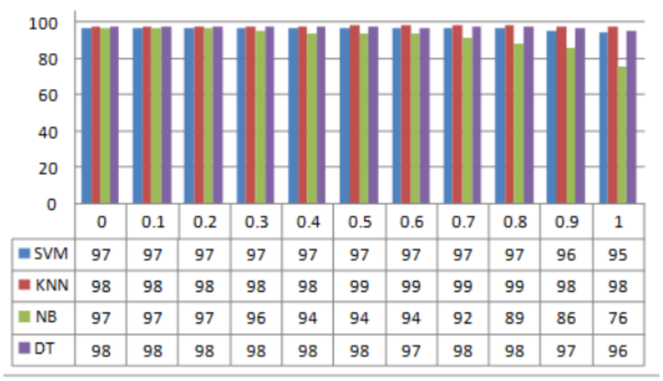
	4	5	6	7	8	9	10
🟦 SVM	75	89	86	88	93	93	95
🟥 KNN	76	89	94	95	97	97	98
🟩 NB	75	75	77	77	78	76	76
🟪 DT	78	91	94	95	96	96	96

Fig. 1. Accuracy of Algorithms vs Maximum Pattern Length

After maximum pattern length (ML)=10 the performances of algorithms remains same.

B. Finding Optimal Value of α

To find optimal value of α we kept maximum pattern length (ML) to 10(obtained from above experiment) and performed five-fold cross validation with different values of α. The α value is changed in range of 0 to 1 with scale 0.1. The performances of classification algorithms are shown in Fig. 2.



	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
🟦 SVM	97	97	97	97	97	97	97	97	97	96	95
🟥 KNN	98	98	98	98	98	99	99	99	99	98	98
🟩 NB	97	97	97	96	94	94	94	92	89	86	76
🟪 DT	98	98	98	98	98	98	97	98	98	97	96

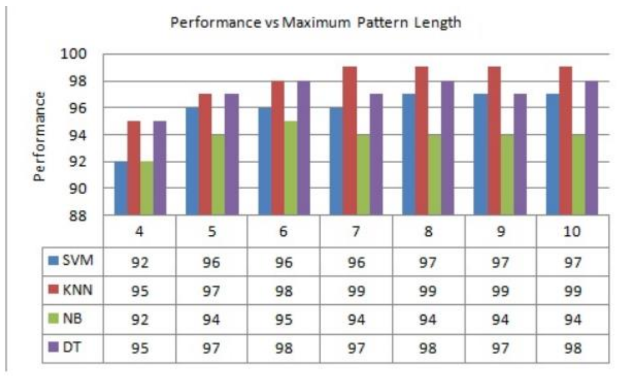
Fig. 2. Accuracy of Alg. Vs α Value when, ML = 10

At value of α=0.5, SVM, KNN and DT gives better performance as compared to other value of α. As value of α increases the performances of SVM, KNN and DT increases and it became maximum at α=0.5 and then it decreases for further value of α. The performance of NB algorithm became maximum at α value 0.2 and then decreases for further value of α.

C. Finding Optimal Value of Maximum Pattern Length

Optimal value of α = 0.5 from Fig. 2, and tried to find the Maximum Pattern Length (ML) which can give better performance. Minimum pattern length is taken as 3 and the maximum pattern length (ML) is increased from 4 to 10.

In Fig. 3, the performances of classification algorithms increases as the maximum pattern length increases and it becomes maximum at length of 8 and remain constant for further values of ML. But, the performance of NB algorithm increases and becomes maximum at length 6 and then decreases for further values of lengths. So, the optimal value of maximum pattern length is selected as 8.



	4	5	6	7	8	9	10
🟦 SVM	92	96	96	96	97	97	97
🟥 KNN	95	97	98	99	99	99	99
🟩 NB	92	94	95	94	94	94	94
🟪 DT	95	97	98	97	98	97	98

Fig. 3. Accuracy of Algorithms vs Maximum Length of Pattern with Constant Value α = 0.5

D. Overall Performance of The Proposed Approach

TABLE IV

PERFORMANCE OF CLASSIFICATION ALGORITHMS.

	Accuracy	Precision	Recall	F-Score
SVM	97	100	95	97
KNN	99	99	99	99
NB	94	100	91	95
DT	98	98	97	97

The optimal value of α is 0.5 and optimal maximum pattern length (ML) is 8. Using these values five-fold cross validation is performed on classification algorithms. The performances of algorithms is shown in TABLE-IV. KNN and DT gives better performances as compared to SVM and NB algorithms. Whereas, the precision of SVM and NB is 100%. The F-Score of KNN is 99% which is higher than other algorithms.

V. CONCLUSION

This work is attempt to detect *Matsar* Sentiment in text. Creation of *Matsar* dataset is done. We proposed a pattern based approach to detect *Matsar* Sentiment in text. This pattern based approach makes use of Pos tags to extract patterns from the text. The proposed algorithm had determined *Matsar* features set which then applied to different classification algorithms and has shown good performances. The accuracy for KNN is 99%, for DT is 98%, for SVM is 97% and for NB is

94%. By increasing dataset size, this approach can cover all kinds of *Matsar* cases.

VI. FUTURE WORK

Future work will be to build large dataset for *Matsar* Sentiment. Building knowledge based lexicon for *Matsar* dataset which would be helpful in detecting *Matsar* Sentiment. Perform various approaches such as, knowledge based approach, rule based approach etc. to detect *Matsar* Sentiment from *Matsar* textual dataset. Selecting different features such as, n-grams, pos tags etc. and combination of these features and compare the results.

VII. REFERENCES

[1]. E. Riloff and J. Wiebe, 2003, "Learning Extraction Patterns for Subjective Expressions" In Conference on Empirical Methods in Natural Language Processing.

[2]. Bo Pang and Lillian Lee, 2004, "A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts" In ACL.

[3]. Bo Pang, S. V., Lillian Lee, 2002, "Thumbs Up?: Sentiment Classifica-tion Using Machine Learning Techniques" In Conference on Empirical Methods in Natural Language Processing-ACL.

[4]. Peter D. Turney, 2002, "Thumbs up or thumbs down?: semantic orien-tation applied to unsupervised classification of reviews" In Association for Computational Linguistics.

[5]. Hongning Wang, Yue Lu, Chengxiang Zhai, 2010, "Latent Aspect Rat-ing Analysis on Review Text Data: A Rating Regression Approach" In International Conference on Knowledge Discovery and Data Mining.

[6]. Ivan Titov, Ryan McDonald, 2008, "A Joint Model of Text and Aspect Ratings for Sentiment Summarization" In ACL-08: HLT.7Benjamin Snyder and Regina Barzilay, 2007, "Multiple Aspect Rank-ing using the Good Grief Algorithm" In Human Language Technolo-gies.

[7]. Kavita Ganesan and ChengXiang Zhai and Jiawei Han, 2010, "Opinosis: A Graph Based Approach to Abstractive Summarization of Highly Redundant Opinions" In 23rd International Conference on Computational Linguistics.

[8]. Giuseppe Di Fabbrizio, Amanda J. Stent, Robert Gaizauskas, 2014, "A Hybrid Approach to Multi-document Summarization of Opinions in Reviews" In 8th International Natural Language Generation Con-ference.

[9]. Jingjing Liu, Yunbo Cao, Chin-Yew Lin, Yalou Huang, Ming Zhou, 2007, "Low-Quality Product Review Detection in Opinion Summa-rization" In Empirical Methods in Natural Language Processing and Computational Natural Language Learning.

[10]. Yang Liu, A. A. X. Y., Xiangji Huang, 2008, "Modeling and Predict-ing the Helpfulness of Online Reviews" InEighth IEEE International Conference on Data Mining.

[11]. Ashwin Rajadesingan, Reza Zafarani, and Huan Liu, 2015, "Sarcasm detection on Twitter: A behavioral modeling approach" In ACM-2015.

[12]. Oren Tsur, Dmitry Davidov, Ari Rappoport, 2010, "ICWSMA great catchy name: Semi-supervised recognition of sarcastic sentences in online product reviews" In Fourth International AAAI Conference on Weblogs and Social Media.

[13]. Andrew B. Goldberg, Nathanael Fillmore, David Andrzejewski Zhiting Xu, Bryan Gibson, Xiaojin Zhu, 2009, "May All Your Wishes Come True: A Study of Wishes and How to Recognize Them" In : The 2009 Annual Conference of the North American Chapter of the ACL.

[14]. J. Ramanand, N. P., Krishna Bhavsar, 2010, "Wishful Thinking Finding sug-gestions and "buy" wishes from product reviews" In Workshop on Computational Approaches to Analysis and Generation of Emo-tion in Text.

[15]. J. Anil Kumar K.M, M. k. M., Asmita Poojari, 2015, "Pattern based Approach for Mining Users Opinion from Kannada Web Documents" In Discovery, 2015.

[16]. J. BOUAZIZI, M., and OTSUKI, T., 2016, "A Pattern-Based Approach for Sarcasm Detection on Twitter" In IEEE-Digital Object Identifier.