Improved Probability Estimator for Human Session Interaction Patterns using Graph Based Algorithm

P.Naga Revathi Manjusha, S.Sailaja
SRK Institute of Technology, Vijayawada
Associate Professor, SRK Institute of Technology, Vijayawada

Abstract

Mining human decision patterns in the meetings or any business interactions are useful to identify the person's opinion within the session. Activities in the session represent the origins of an individual and mining methods assist to analyze how person delivers their opinion at different paths. In this proposed work, meeting interactions are classified as comment, propose, request-information, acknowledgement, ask, positive and negative opinion. Traditional human interaction techniques are used to detect and analyze patterns to find various types of new knowledge on interactions. Traditional methods fail to extract sub frequent patterns in the interaction flow. Human interaction flow is represented as graph along with their opinions. Graph based pattern mining algorithm was planned to extract relevant patterns from the meeting interaction dataset. Proposed work has extended to extract interaction patterns using DAG (Directed Acyclic Graph) based mining algorithm. Graph-based Substructure mining algorithm which discovers the frequent substructure paths from the candidate patterns of DAG algorithm. Substructures help to predict the probability of associate patterns within the session. Proposed algorithm efficiently uses different support and confident measures to extract user interaction patterns. Experimental study shows that the proposed model extract high relevant interaction patterns with less time and high accuracy.

Keywords – *Human Meeting*, Decision Patterns, Support,Substructure.

I.INTRODUCTION

Face-to-face conversation is probably the most basic kind of communication in our own life and is utilized for sharing information, understanding others' emotion/ intention, and making decisions. To enhance our communication capability beyond conversations immediately, the automatic study of a conversation scene is a basic technical requisite to enable effective teleconferencing, archiving/summarizing meetings, and to realize communication via social agents and robots. Within the face-to-face setting, the messages include simply not only verbal but additionally nonverbal messages. As such, meetings contain a lots of rich business information that really is often not formally documented. Capturing all of these issues informal meeting information has long been topic of to pump several communities within the last decade. Some of the most common technique to capture meeting info is through note-taking. However, fully noting your content regarding a meeting is a hard job, and can bring about an inability to both take notes and participate in the meeting. The possible benefits to owning a meeting record to start with and of course the troubles with traditional meeting recording then again have triggered using technology to construct meeting records. While technology automatically captures meeting activities, humans are left liberated to actively keep up with discussions and synthesize what is wrong and what not around them, without worrying about tediously preserving details for later memory. The tactic of one's choice for recording meetings has been audio and video, which can certainly provide a comprehensive meeting record that lets people to see who was present and what was discussed[1-4].

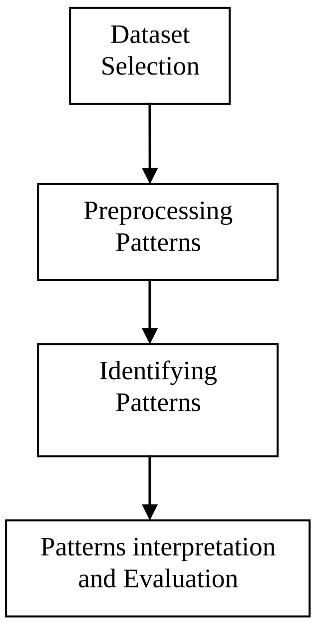

Fig.1. Basic Pattern discovery Process

Fig.1. shows the basic flowchart of pattern

discovery in human interaction process. In this process dataset is preprocessed before applying the pattern discovery process. Finally patterns are interpreted and evaluated for decision making.
Within the social dynamics, an example would be human interaction will be the one of the important for understanding how a human's behavior or human activities underneath of the meeting and determining whether the meeting was well organized or possibly not will be the securely at the top of all lists of issues inside the meetings. Several methods could have been designed to obtain the interaction of the flow in the meeting in every human. To further know about human and interference of one's human interactions in meetings, here desired to uncover advanced level semantic knowledge which can include which interactions flow often appear in legal representative ,what interaction flow discussion usually follows, and relationships connecting the exist among interactions. This data will help to outline important patterns of interaction. Meetings constitute the characteristic and important cases within the people interaction, becomes challenging problem for several conditions along with a relatively well-defined dictionary of relevant events.

Opinion mining is basically a kind of natural language processing for tracking the sense of the general public with regards to a particular product. The sentiment found within comments, feedback or critiques provide useful indicators for many different purposes. These sentiments can be categorized either into two categories: either positive or negative; or into an point scale, e.g., , good, satisfactory, bad, worse. A sentiment analysis task works extremely well for being classification task where each category displays sentiment.

An interaction flow discuss about session with triggering relationship connecting the groups. Everyday is basically a unit involved with a gathering that starts off by using a interaction and concludes which includes an interaction. Here, spontaneous interactions are those people who are initiated by the person spontaneously and reactive interactions are triggered for another interaction. By way of example, Opinion is in most instances spontaneous interactions. Session interactions aren't just based upon its type, but interaction type for nodes labeled on their annotator manually. Hence, a session already has few interactions. A meeting discussion involves a sequence of sessions, wherein participants discuss topics continuously.

II.LITERATURE SURVEY

TREEMINER is a robust algorithm which discover all frequent patterns within a tree, using a data structure called scope-list. In this particular system a pattern matching tree mining algorithm was proposed to extract relevant patterns. After performing these steps, several experiments were conducted test the performance of system and scalability of them methods and discover that pattern matching approach performs quicker than another system; plus it has good raise properties. Plus it possesses an applying tree mining to evaluate real web logs for usage patterns.

In earlier work Chopper [3] and XSpanner systematically develop the two algorithm pattern growth techniques for mining frequent tree patterns. Experimental results show that the newly developed algorithms outperform TreeMinerV. Moreover XSpanner is significantly a lot faster then Chopper in many different cases. The procedure for frequent tree pattern mining is efficient and scalable when the patterns aren't too complex. Regardless of the fact that there are several complex patterns inside the data set.

Several works has been proposed to discover Human behavior patterns through the use of stochastic techniques. Magnusson [3] proposed a pattern detection method, called Tree-pattern to find hidden time patterns in human behavior. Tree-pattern has also been adopted in various applications for instance interaction analysis and sports research.

Casas-Garriga[4] proposed unfixed window interval timeline to extract patterns within the range. They used episodes i.e., collections of events occurring together.

Morita et al. proposed a pattern based method to interpret human interactions within the given threshold. Sawamoto et al. proposed a technique for representing relevant decision patterns in medical interviews.

III. PROPOSED SYSTEM

The meaning of human interaction meeting varies depending on the purpose and types of the session meeting. In this proposed approach, four different types of user's feedback or opinions are considered namely comment, propose, acknowledgement, request, ask, positive and negative opinions. In this work, comment is tagged as com, propose as pro, request as req, ask as ask, positive as pos, negative as neg. Each tag can be used to represent session class label. Human Interaction flows may vary from data to data and size of the dataset. Dataset is prepared from one of the session in human interaction meeting.

Human Interaction Graph Representation

Human interaction diagram is represented in the form of Graph as shown in Fig.2. An interaction flow is a list of interactions of the participants along with sessional relationship between them. Each participant has their own feedback towards the discussion topic within the session. Each node in the graph denotes the opinion tag of the participant. Large number of initial paths from the root to the end of the node are given to the proposed algorithm.

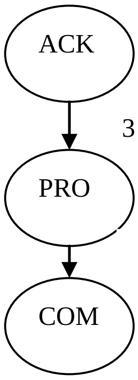

Fig. 2 (a) Weighted Single Path Human Interaction Diagram

In the Fig.2 (a) Weighted single path human interaction diagram, ACK-PRO-COM path indicates the new session feedback of the participants. Size of the path depends on the number of opinions of the participants.

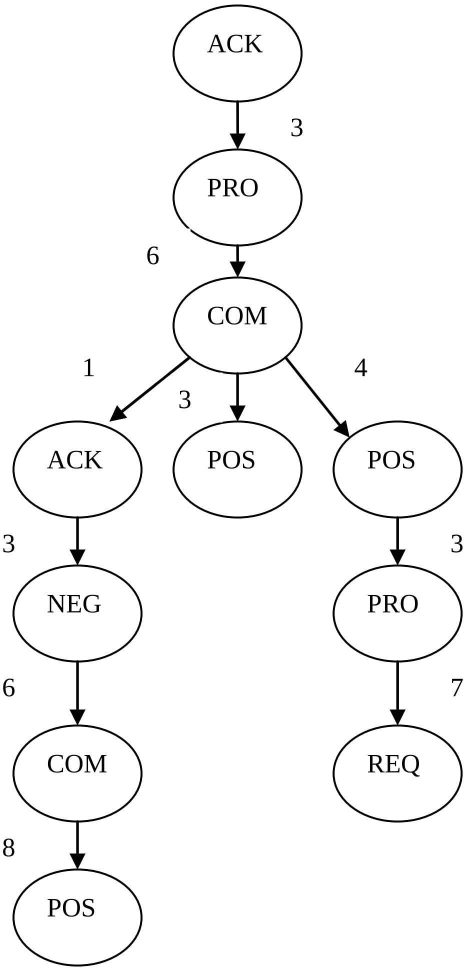

Fig. 2 (b) Weighted Multi-Path Human Interaction Diagram

In the Fig.2 (b) Weighted Multi-path human interaction diagram, ACK-PRO-COM-POS, ACK-PRO-COM-ACK-NEG-COM-POS and ACK-PRO-COM-POS-PRO-REQ are the different paths which indicates the new session feedback of the participants. Size of the paths depends on the number of opinions of the participants and session time.

Proposed Workflow

In this proposed work, human interaction meeting dataset is used with the specified tags in the given data format[1].
Data format : tag1tag2:count
Example: PROACK:5,ACKNEG:2

Fig 3. Shows the proposed workflow structure which consists of following steps:

Step1: Loading Human Interaction Meeting Dataset.
Step 2: Identifying the data structure in the dataset.
Data format: tag1tag2:count
Step 3: Applying proposed Graph Mining Algorithm.
Step 4: Dynamic support and confident measures are used to filter patterns from the graph mining algorithm.
Step 5 : Filtering sub patterns using thresholds.
Step 6: Final Decision making human interaction patterns.

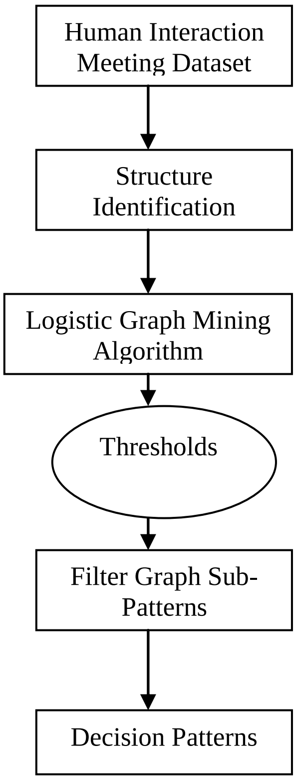

Fig 3. Proposed WorkFlow

Weighted Graph Mining Algorithm:

Input : G(DB,minsup)
DB Format:V1V2W

Procedure:
ReadInput(file)
String[] input=Each Record.(ex:AE5)
For(int i=0;I,inputsize;i++)
Do
Vertex v1=getFirstNode as input[0];
Vertex v2=getSecondNode as input[1];
Weight w=getThirdValue as input[2];
Create DirectedEdge(v1,v2,w);

Subprocedure 1:
Add edge to DAG graph as
{
U=from node;
V=to node;
LinkedHashSet<DirectedEdge> adj_edges_from = edges_from_map.get(v);
LinkedHashSet<DirectedEdge> adj_edges_to = edges_to_map.get(w);
If(edgelistfrom==null)
then
edges_from_map.put(v, adj_edges_from); // create a new edge_adj_edges_from.add(e); // add this edge to the already existing set
if (adj_edges_to == null) // empty set of edges to vertex v
{
adj_edges_to = new LinkedHashSet<DirectedEdge>();
edges_to_map.put(w, adj_edges_to);
}
_edges_to.add(e); // add this edge to the already existing set
num_edges += 1;
level++;
}
End for
String st[]={"Map-A"," Map-B’"," Map--C’"," Map--D’"," Map--E’"," Map--F’"," Map--G’"," Map--H’"," Map--I’"};
String st1[]={"Map-A"," Map-B’"," Map--C’"," Map--D’"," Map--E’"," Map--F’"," Map--G’"," Map--H’"," Map--I’"};
for(int i=0;i<8;i++)
do
for(int j=i+1;j<9;j++)
do
src=st[i];
dest=st[j];
connects= new GetFrequentAllSubDAGs1 (graph, src, dest, val);

done//inner loop ends
done

Subprocedure2:

GetFrequentAllSubDAGs1(MeetDigraph G, String source, String dest, int val1)
{

// queue of f that have been traversed so far
LinkedList<String> f = new LinkedList<String>();

```
            f.add(source);
            GetDAGPaths(G, f);
        }
```

Subprocedure 3:

```
GetDAGPaths(MeetDigraph G,
LinkedList<String> tempF)
    {
     //  Get all SubDAG paths from G

        String
st[]={"PRO","ACK","NEG","POS","COM","REQ"
,"ACC","ASK","REJ"};
        ArrayList pat;
        String sb=null;
        Character array ch[]=curr_path// get
SubDag path
        int le=0;
 while(le<(ch.length))
 {

          if(le==ch.length-1)
          then
          if(ch[le]=='A')
              {
                pat.add("->PRO");
                sb.append("A-");
              }
          else if(ch[le]=='B')
              {
              pat.add("->ACK");
               sb.append("B-");
              }………
         End if
          getSupportCount for each SubDAG
rule;
          calculate minsup for each SubDAG:

minsuport=(supDAGs*avgweight)/(totDAGs*Meet
Digraph.maxweight);
 if(minsuport>minsup)
 then
Display("SUPPORT SATISFIED is "+minsuport);
Display("Satisfied SubDAG rule");
End if
}//end procedure
}
```

IV. EXPERIMENTAL RESULTS

All experiments are performed with the configurations Intel(R) Core(TM)2 CPU 2.13GHz, 2 GB RAM, and the operating system platform is Microsoft Windows XP Professional (SP2). This framework requires third party libraries junit,jama.

SUPER DAG :

```
41 41
D: D->E 6  D->F 5  D->C 4  D->H 6
E: E->B 3  E->D 3  E->A 3
F: F->E 5
G: G->C 6  G->C 6  G->H 2  G->F 3
A: A->B 5  A->D 5  A->E 3  A->G 4  A->H 4  A-
>G 2  A->F 3  A->C 5  A->I 5  A->H 3
B: B->C 4  B->G 2  B->H 2  B->I 5  B->C 8  B->G
2  B->H 4  B->I 2
C: C->D 8  C->E 2  C->I 3  C->D 8
H: H->I 3  H->A 2
I: I->E 2  I->G 4  I->C 3  I->G 5  I->H 3
--------------------------
SUPPORT SATISFIED is 0.016369047619047 62
Rule is PRO-ACC-NEG--->COM
SUB-DAG 37: [PRO-, ACC-, NEG-, -->COM]
Without Duplicate [PRO-, ACC-, -->COM, NEG-]
path is A-G-C-E
Total Logistic Regression 4.7321383341814505
Total Weight of DAG :33 Avg weight is
4.714285714285714
Max weight is 8.0
Minimum support Threshold
0.015926640926640926
--------------------------
SUPPORT SATISFIED is 0.015926640926640926
Rule is PRO-ACC-NEG--->COM
SUB-DAG 38: [PRO-, ACC-, NEG-, -->COM]
Without Duplicate [PRO-, ACC-, -->COM, NEG-]
path is A-G-C-E
Total Logistic Regression 4.7321383341814505
Total Weight of DAG :33 Avg weight is
4.714285714285714
Max weight is 8.0
Minimum support Threshold
0.015507518796992482
--------------------------
SUPPORT SATISFIED is 0.015507518796992482
Rule is PRO-ACC-NEG--->COM
SUB-DAG 39: [PRO-, ACC-, NEG-, -->COM]
Without Duplicate [PRO-, ACC-, -->COM, NEG-]
path is A-G-C-E
Total Logistic Regression 4.7321383341814505
Total Weight of DAG :33 Avg weight is
4.714285714285714
Max weight is 8.0
Minimum support Threshold
0.01510989010989011
--------------------------
SUPPORT SATISFIED is 0.01510989010989011
Rule is PRO-ACC-NEG--->COM
SUB-DAG 40: [PRO-, ACC-, REQ-, -->COM]
Without Duplicate [PRO-, ACC-, REQ-, -->COM]
path is A-G-F-E
Total Logistic Regression 2.968830682950186
Total Weight of DAG :20 Avg weight is
2.857142857142857
Max weight is 8.0
Minimum support Threshold
0.008928571428571428
```

SUB-DAG 41: [PRO-, ACC-, NEG-, -->COM]
Without Duplicate [PRO-, ACC-, -->COM, NEG-]
path is A-G-C-E
Total Logistic Regression 4.7321383341814505
Total Weight of DAG :33 Avg weight is 4.714285714285714
Max weight is 8.0
Minimum support Threshold 0.014372822299651568

SUPPORT SATISFIED is 0.014372822299651568
Rule is PRO-ACC-NEG--->COM
SUB-DAG 42: [PRO-, ACC-, NEG-, -->COM]
Without Duplicate [PRO-, ACC-, -->COM, NEG-]
path is A-G-C-E
Total Logistic Regression 4.7321383341814505
Total Weight of DAG :33 Avg weight is 4.714285714285714
Max weight is 8.0
Minimum support Threshold 0.01403061224489796

SUPPORT SATISFIED is 0.01403061224489796
Rule is PRO-ACC-NEG--->COM
SUB-DAG 43: [PRO-, ASK-, REJ-, -->COM]
Without Duplicate [ASK-, PRO-, -->COM, REJ-]
path is A-H-I-E
Total Logistic Regression 3.0971747031474837
Total Weight of DAG :21 Avg weight is 3.0
Max weight is 8.0
Minimum support Threshold 0.00872093023255814

SUB-DAG 44: [PRO-, ASK-, REJ-, -->COM]
Without Duplicate [ASK-, PRO-, -->COM, REJ-]
path is A-H-I-E
Total Logistic Regression 3.0971747031474837
Total Weight of DAG :21 Avg weight is 3.0
Max weight is 8.0
Minimum support Threshold 0.008522727272727272

SUB-DAG 45: [PRO-, ASK-, PRO-, -->COM]
Without Duplicate [ASK-, PRO-, -->COM]
path is A-H-A-E
Total Logistic Regression 2.968830682950186
Total Weight of DAG :20 Avg weight is 2.857142857142857
Max weight is 8.0
Minimum support Threshold 0.007936507936507936

SUB-DAG 46: [PRO-, ASK-, PRO-, -->COM]
Without Duplicate [ASK-, PRO-, -->COM]
path is A-H-A-E
Total Logistic Regression 2.968830682950186
Total Weight of DAG :20 Avg weight is 2.857142857142857
Max weight is 8.0
Minimum support Threshold 0.007763975155279503

SUB-DAG 47: [PRO-, ASK-, PRO-, -->COM]
Without Duplicate [ASK-, PRO-, -->COM]
path is A-H-A-E
Total Logistic Regression 2.968830682950186
Total Weight of DAG :20 Avg weight is 2.857142857142857
Max weight is 8.0
Minimum support Threshold 0.007598784194528876

SUB-DAG 48: [PRO-, ASK-, REJ-, -->COM]
Without Duplicate [ASK-, PRO-, -->COM, REJ-]
path is A-H-I-E
Total Logistic Regression 3.0971747031474837
Total Weight of DAG :21 Avg weight is 3.0
Max weight is 8.0
Minimum support Threshold 0.0078125

SUB-DAG 49: [PRO-, ASK-, REJ-, -->COM]
Without Duplicate [ASK-, PRO-, -->COM, REJ-]
path is A-H-I-E
Total Logistic Regression 3.0971747031474837
Total Weight of DAG :21 Avg weight is 3.0
Max weight is 8.0
Minimum support Threshold 0.007653061224489796

SUB-DAG 50: [PRO-, ACC-, NEG-, -->COM]
Without Duplicate [PRO-, ACC-, -->COM, NEG-]
path is A-G-C-E
Total Logistic Regression 4.7321383341814505
Total Weight of DAG :33 Avg weight is 4.714285714285714
Max weight is 8.0
Minimum support Threshold 0.011785714285714287

SUPPORT SATISFIED is 0.011785714285714287
Rule is PRO-ACC-NEG--->COM
SUB-DAG 51: [PRO-, ACC-, NEG-, -->COM]
Without Duplicate [PRO-, ACC-, -->COM, NEG-]
path is A-G-C-E
Total Logistic Regression 4.7321383341814505
Total Weight of DAG :33 Avg weight is 4.714285714285714
Max weight is 8.0
Minimum support Threshold 0.011554621848739496

SUPPORT SATISFIED is 0.011554621848739496
Rule is PRO-ACC-NEG--->COM
SUB-DAG 52: [PRO-, ACC-, NEG-, -->COM]
Without Duplicate [PRO-, ACC-, -->COM, NEG-]
path is A-G-C-E
Total Logistic Regression 4.7321383341814505
Total Weight of DAG :33 Avg weight is 4.714285714285714

Max weight is 8.0
Minimum support Threshold
0.011332417582417582

SUPPORT SATISFIED is 0.011332417582417582
Rule is PRO-ACC-NEG--->COM
SUB-DAG 53: [PRO-, ACC-, NEG-, -->COM]
Without Duplicate [PRO-, ACC-, -->COM, NEG-]
path is A-G-C-E
Total Logistic Regression 4.7321383341814505
Total Weight of DAG :33 Avg weight is
4.714285714285714
Max weight is 8.0
Minimum support Threshold
0.011118598382749326

SUPPORT SATISFIED is 0.011118598382749326
Rule is PRO-ACC-NEG--->COM
SUB-DAG 54: [PRO-, ACC-, REQ-, -->COM]
Without Duplicate [PRO-, ACC-, REQ-, -->COM]
path is A-G-F-E
Total Logistic Regression 2.968830682950186
Total Weight of DAG :20 Avg weight is
2.857142857142857
Max weight is 8.0
Minimum support Threshold
0.006613756613756614

SUB-DAG 55: [PRO-, ACC-, NEG-, -->COM]
Without Duplicate [PRO-, ACC-, -->COM, NEG-]
path is A-G-C-E
Total Logistic Regression 4.7321383341814505
Total Weight of DAG :33 Avg weight is
4.714285714285714
Max weight is 8.0
Minimum support Threshold
0.010714285714285714

SUPPORT SATISFIED is 0.010714285714285714
Rule is PRO-ACC-NEG--->COM
SUB-DAG 56: [PRO-, ACC-, NEG-, -->COM]
Without Duplicate [PRO-, ACC-, -->COM, NEG-]
path is A-G-C-E
Total Logistic Regression 4.7321383341814505
Total Weight of DAG :33 Avg weight is
4.714285714285714
Max weight is 8.0
Minimum support Threshold
0.010522959183673469

Total rules are 2
Number of SubDAG'S are = 25

Accuracy Comparison:

Threshold(/100)	Number of Patterns
1	21
2	13
3	9
4	6
5	5
6	4

Table 1. Threshold Vs Number of Patterns

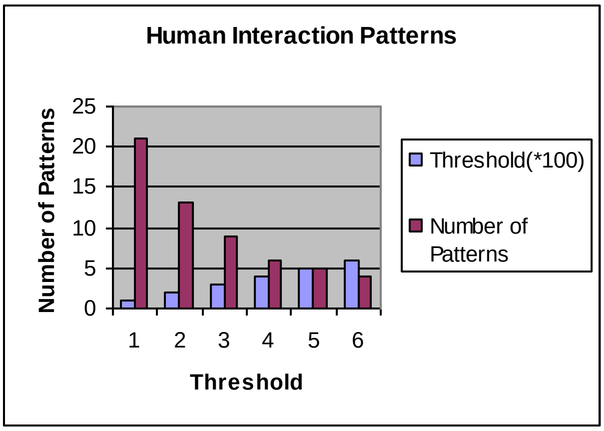

Fig 4. Comparison Between Threshold Vs Patterns

Threshold(*100)	Number of Patterns	Time(s)
1	21	4
2	13	5
3	9	5
4	6	5.2
5	5	4.9
6	4	5

Table 2: Time Vs Patterns

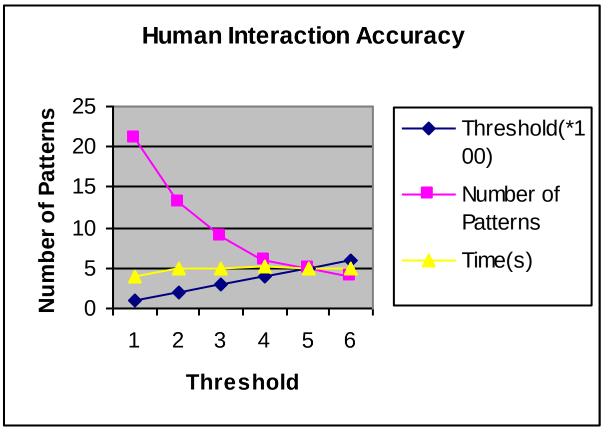

Fig. 5. Time Vs Number of Patterns

V. CONCLUSION AND FUTURE SCOPE

This paper proposes a new weighted based graph mining algorithm to extract relevant high patterns from the human interaction dataset. Proposed approach uses DAG algorithm to find sub patterns within the structure. Each DAG path has total weight used to predict the probability of the path within the graph structure. Graph-based Substructure mining algorithm which discovered the frequent substructure paths from the candidate patterns of DAG algorithm. Substructures help to assume the probability of another type of interaction within the session. Proposed algorithm efficiently uses different support and confident measures to extract user interaction patterns. Experimental study shows that the proposed model extract high relevant interaction patterns with less time and high accuracy.

VI REFERENCES

[1] Mining Human Decision Patterns Using Weighted Substructure DAG algorithm, K.Aparna and k .Venkataraju, International Journal of Applied Engineering Research (IJAER).

[2] HIDDEN INTERACTION PATTERN DISCOVERY OF HUMAN INTERACTION IN MEETINGS,.

[3] DETECTING INTEREST LEVEL PATTERNS OF HUMAN INTERACTION USING TREE BASED MINING, Sahaya Sachithananthi Yesuraj , C. Balasubramanian, International Journal of Emerging Technology and Advanced Engineering, Volume 3, Special Issue 1, January 2013

[4] An Efficient Interaction Pattern Discovery For Human Meetings, A.Nandha Kumar , N.Baskar, International Journal of Computer Trends and Technology (IJCTT) - volume4 Issue5–May 2013.

[5] Modeling Human-Agent Interaction with Active Ontologies, Didier Guzzoni and Charles Baur, Adam Cheyer.

[6] Discovering Patterns in Interactions between Humans and Animals by Using Tree Based Mining, Palivela Hemant, VijayKumar S, Sharadha K A, International Journal of Engineering Research & Technology.

[7] OPINION MINING AND ANALYSIS: A SURVEY, Arti Buche, International Journal on Natural Language Computing (IJNLC) Vol. 2, No.3, June 2013.

[8] Classification of Opinion Mining Techniques, Nidhi MishraInternational Journal of Computer Applications (0975 – 8887) Volume 56– No.13, October 2012.

[9] METHODOLOGICAL STUDY OF OPINION MINING AND SENTIMENT ANALYSIS TECHNIQUES, Pravesh Kumar Singh, International Journal on Soft Computing (IJSC) Vol. 5, No. 1, February 2014