Annals of the Academy of Romanian Scientists
Series on Science and Technology of Information
ISSN 2066 - 8562 Volume **2**, Number **1**/**2009**

CH.A.M.P. – A PROGRAM FOR CHAT MODELING AND ASSESSMENT

Mihai DASCĂLU[1,2], Stefan TRĂUȘAN-MATU[1]

Rezumat. *Lucrarea propune o metodă şi un sistem implementat de evaluare a competenţelor participanţilor din cadrul unui mediu colaborativ de tip chat. În cadrul mecanismului de notare au fost luate in calcul metrici specifice reţelelor sociale, au fost folosite tehnici de text-mining, prelucrarea limbajului natural şi analiză semantică latentă (LSA – Latent Semantic Analysis). Modelul pentru interacţiunea între participanţi, evoluţia şi notarea lor joacă un rol important în vizualizarea rezultatelor analizei. Un alt sistem a fost dezvoltat pentru a permite evaluarea manuală a fiecărui chat în vederea obţinerii unui corpus de referință (”golden standard”) şi în vederea învăţării din corpus folosind LSA şi Wordnet.*

Abstract. *The paper describes a method and an implemented system used for evaluating participants' competencies in a chat collaborative environment. The assessment provides a grading mechanism based upon social network metrics, text mining, natural language pragmatics and latent semantic analysis. The model for participant interaction, evolution and grading plays an important role in the visualization of the analysis results. Another system has been developed in order to manually evaluate each chat and obtain the "golden standard" and learn from the corpus using LSA and WordNet.*

Keywords: Computer-Supported Collaborative Learning, chat, polyphony, evaluation, annotation, social networks, semantic web, Latent Semantic Analysis

1. Introduction

As the web evolved into a social environment, other communication channels were developed allowing users to exchange ideas, thoughts and information worldwide.

In this context instant messaging and forums emerged becoming a viable alternative to classic learning: Computer Supported Collaborative Learning [7, 9]. However, new difficulties involving manual chat analysis appeared because of the large amount of information and an automatic system's help would be required.

For example, a professor's evaluation is an extremely time consuming process and social networks and natural language processing would be helpful.

[1]Faculty of Automatic Control and Computers, "Politehnica" University of Bucharest, Romania. Correspondent member of the Academy of Romanian Scientists.
[2]CCT S.R.L, Bucharest, Romania.

The paper presents and evaluates an automatic assessment system by comparing results with the ones obtained from manual evaluation. The inputs for the system are the utterances, their sequencing and the explicit links. Based upon these inputs, the system builds the social network using several metrics, ranging from the simplest ones like the dimension of utterances to more sophisticated ones as user ranking and assigns a grade to each participant [2]. Each utterance is evaluated using Latent Semantic Analysis (LSA, [5]) and part of speech analysis; the previous utterance and a set of predefined keywords are also taken into consideration.

The second section is focused on the analysis factors commonly used in socials networks, the evaluation system and generated graphics. The third part of the paper presents LSA and its use in the program, followed by an overall view of the system's accuracy in grading participants.

2. Analysis factors and the evaluation system

For the evaluation process a set of metric have been computed, starting from the simplest feature – the number of characters written by a participants, and ending with a user rank algorithm. But information like the number of characters or the average number of characters per utterance offers only a raw base for analysis, quality being more important than quantity. Therefore, in order to obtain a better efficacy, a balance between the length of the interventions and the information held within must be achieved.

Moreover, for a social analysis of the chat, social factors are taken into account. Consequently, a graph in which nodes are participants in a collaborative environment is generated from explicit links (obtained from the explicit referencing facility of the chat environment used [4]) between utterances and implicit ones obtained using natural language processing techniques (for example, [8]) – in this case LSA.

From graph theory, the first two measures taken into consideration are in-degree (the number of arcs entering a node) and out-degree (the number of outgoing arcs from a specified node. Considering the social environment three types of centralities are identified: closeness, graph centrality and eigenvector. Closeness evaluates the centrality proportional with the inverse of the minimal distance between the current node and all other nodes. Graph centrality is a relative closeness by evaluating the greatest distance between the considered node and all other nodes. The Floyd-Warshall algorithm can be used because it provides the shortest distance for each pair of nodes in $O(n^3)$ complexity [3]. The eigen-value approach attaches a relative mark to every node following the following principle: a connection to a higher ranking node is more important than a set of connections to inferior ranked nodes [2]. For participant assessment the following assumptions are made:

- For all negative values, the absolute values are considered;
- For both positive and negative values, the percentage is distributed between the highest and smallest values;
- For all positive values, the percentages are calculated using a scale from 0 to the maximum value.

Another metric for social network is user rank, based on the Google Page Rank algorithm [10]. A user's rank is influenced by the other participants' ranks that are directly addressing him. Therefore, the utterances the user receives and the rank of the participants he is talking to are the main factors that determine his current ranking. The system uses an iterative method based on this equation:

$$UR(A) = (1-d) + d\left(\frac{UR(t_1)}{c(t_1)} + ... + \frac{UR(t_n)}{c(t_n)}\right) \quad (1)$$

where UR= user rank; $c(t_i)$ = number of utterances exchanged between user t_i and user A; d=a constant (in the implementation 0.85), used for a faster convergence of the system.

A serious problem encountered in a chat environment is determined by the high occurrence level of misspelled words, abbreviations and emoticons. For handling these sorts of issues, besides using a list of stop words to eliminate irrelevant parts of an utterance, Jazzy library [11] has been used for spellchecking, with a few modifications. Besides trying to add a space in a word and check if the overall Levenshtein distance is smaller, the occurrence matrix of words and LSA have been used to enhance Jazzy: similarity with other words which determine the context of a specific misspelled word are taken into consideration. Furthermore, spellchecking is double-checked using WordNet as a dictionary.

For stemming, Snowball [12] was chosen because, in the context of prior usage of other stemmers as Porter [13] or Lovin [14], it offered better results. Moreover, in the Porter's web page Snowball is recommended.

Two kind of evaluations based on the above mentioned social network factors are computed:

- A quantity based approach where the number of exchanged utterances between participants is taken into consideration.
- A qualitative point of view where each utterance is graded where several factor besides the length are used:
    - the number of key words which remain after eliminating stop words, spell-checking and stemming - MClength.
    - the number of occurrences of a word and their relevance to a set of

keywords: no_occurences.

- the level at which the current utterance is situated in the thread.

The following formulas are obtained:

$$mark_{empiric} = \frac{\frac{length}{6} + \frac{MClength * 5}{6}}{1 + \frac{1}{level}} * \left(1 + \frac{1}{10} * relevance\right), \tag{2}$$

$$relevance = \sum_k \ln(no\_occurances_k + 1) * Sim(word_k, list\_of\_keywords),$$

where relevance is computed for all words that remain after initial processing of an utterance and similarity will be presented in the next section.

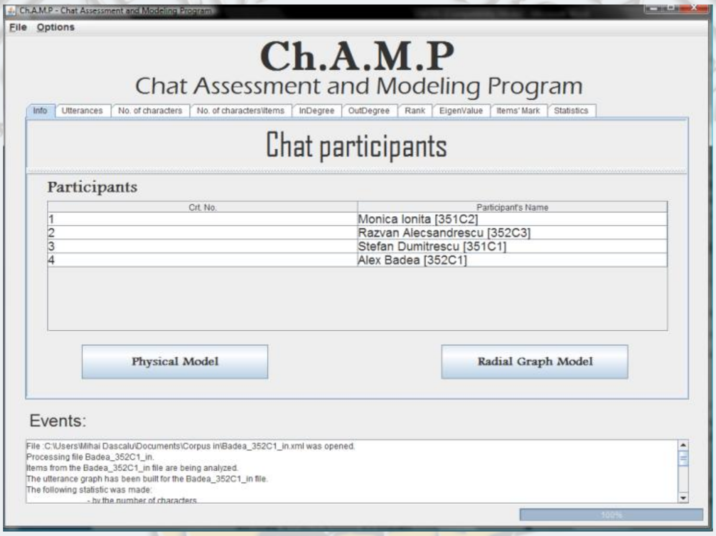

Fig. 1. The ChAMP Main Interface

Based on this empiric mark, the final grade of an utterance is obtained:

$$mark_{final} = mark_{previous\_utterance} + coefficient * mark_{empiric}, \tag{3}$$

where the coefficient is determined from the type of the current and previous (linked with) utterance. For the coefficient determination, identification of speech acts plays an important role: verbs, punctuation signs and certain keywords are inspected. In the current implementation, utterances are grouped in: negations, confirmations, questions and affirmations and the coefficient values are obtained from a predefined matrix.

For determining the final grade of a participant, all these factors applied both on the quantity / quality measurements are given a weight:

$$final\_grade_i = \sum_k weight_k * grade\_factor_{k,i}, \tag{4}$$

where *i* is the current participant and *k* is the measure taken into consideration.

For the social network visualization two models were created based upon the Prefuse framework [15]:

- a physical driven model – a participant is considered a planet, it has his own mass, the length between users is based on utterance marks exchanged and the elasticity coefficients are also modified in order to obtain a more realistic model of the network;

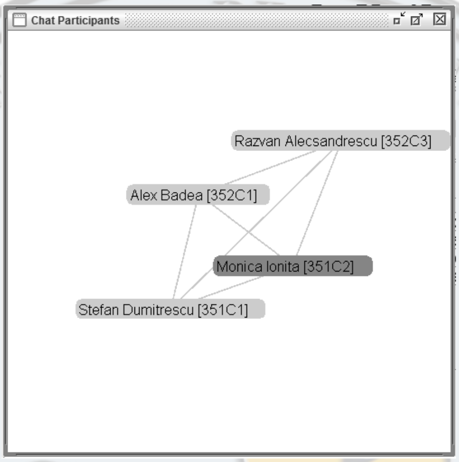

Fig. 2. A physical based model approach for visualizing the social network

- a radial model which offers a central perspective – the graph is focused on the central participant and his neighbors; the view can be observed from the perspective of any user, plus it offers search capabilities useful in larger social networks.

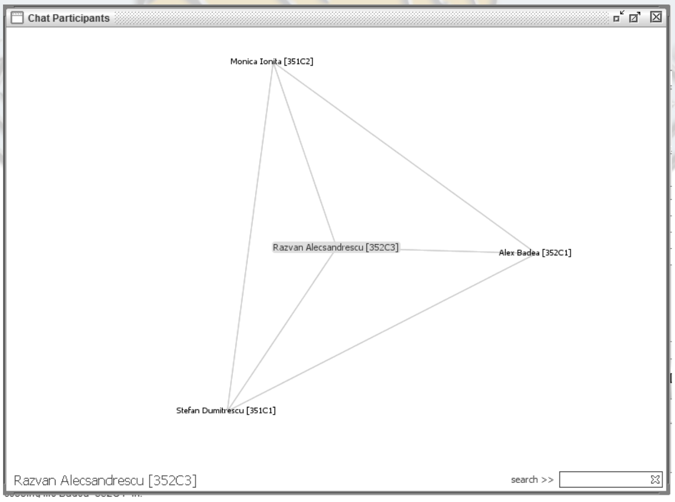

Fig. 3. A radial representation of the social network, including participant search

For each social network factor and the final statics a bar chart is generated for better visualization and understanding of user ratings [1].

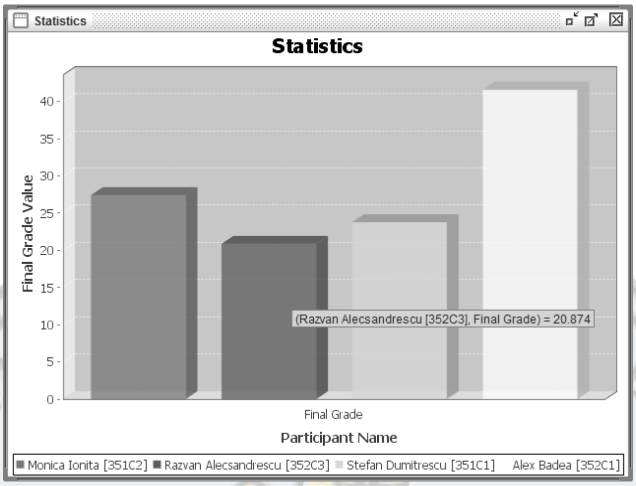

Fig. 4. The generated graph for each factor (including the number of utterances / utterance grades and final participant statistics)

On the other hand, the system offers the possibility to view the overall chat evolution based on each utterance's final grade. The grade of the discussion will be influenced by each utterance, thus depending on the type and speech acts of the current utterance, negative values are possible [1]. Also, visualization of a single thread based on the first utterance of interest is possible.

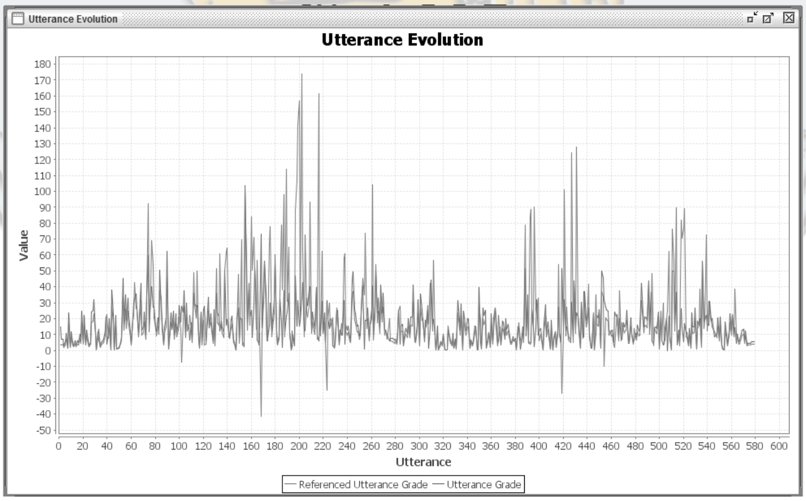

Fig. 5. Generated graph representing utterance evolution in the whole chat

Similar with utterance evolution, visualization of each participant's evolution is possible by calculating for a specific utterance in the chat the overall contribution so far of a particular participant [2].

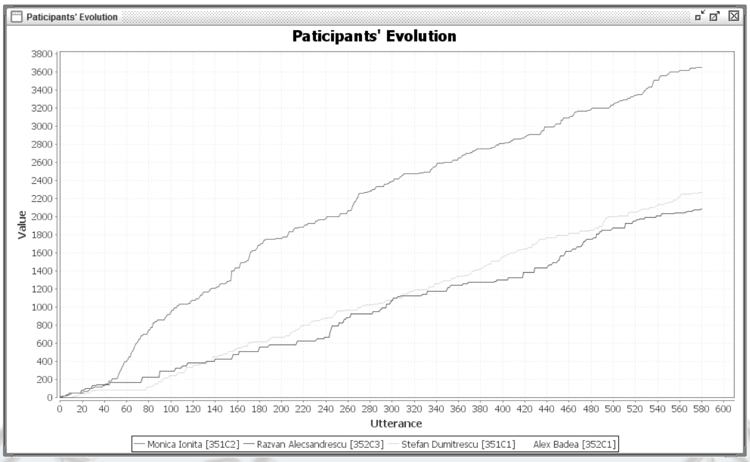

Fig. 6. Generated graph representing overall participant evolution in a chat

An important feature in the overall system evaluation is the manual annotation module which allows comprehensive corpus annotation in the teaching process. This system allows the following facilities:

- import chats from HTML and save them as XML;
- add annotations to utterances, participants (for each utterance, sequence of 20 utterances or overall for the entire chat) or intense collaboration zones;
- topics identification following the overall chat evolution;
- implicit links markup allowing reference type and pattern identification.

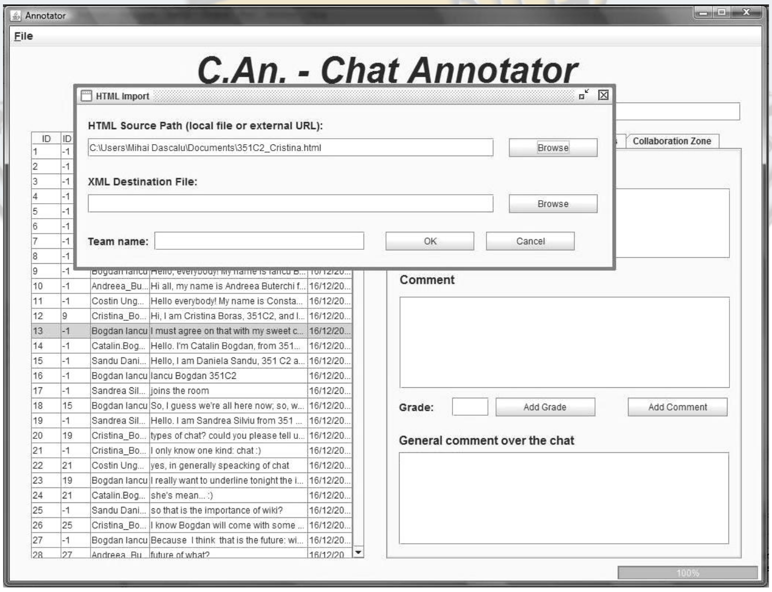

Fig. 7. Main interface for the chat annotator module

3. Latent Semantic Analysis (LSA)

3.1. General Description

LSA is a technique used in natural language processing, in particular in vector-space based semantics, used for analyzing relationships between a set of documents and the contained terms by projecting the terms in sets of concepts related to the documents.

LSA uses a term-document bi-dimensional array which describes the occurrence of term in documents. It is a sparse array whose rows typically correspond to stem words which appear in documents (which are the columns of the array).

LSA transforms the occurrence array into a relation between terms and concepts, and a relation between those concepts and the used documents. Thus the terms and the documents are now indirectly related through concepts [5]. This transformation is obtained by a singular-value decomposition of the array and a reduction of its dimensionality, similarly with the least-squares method.

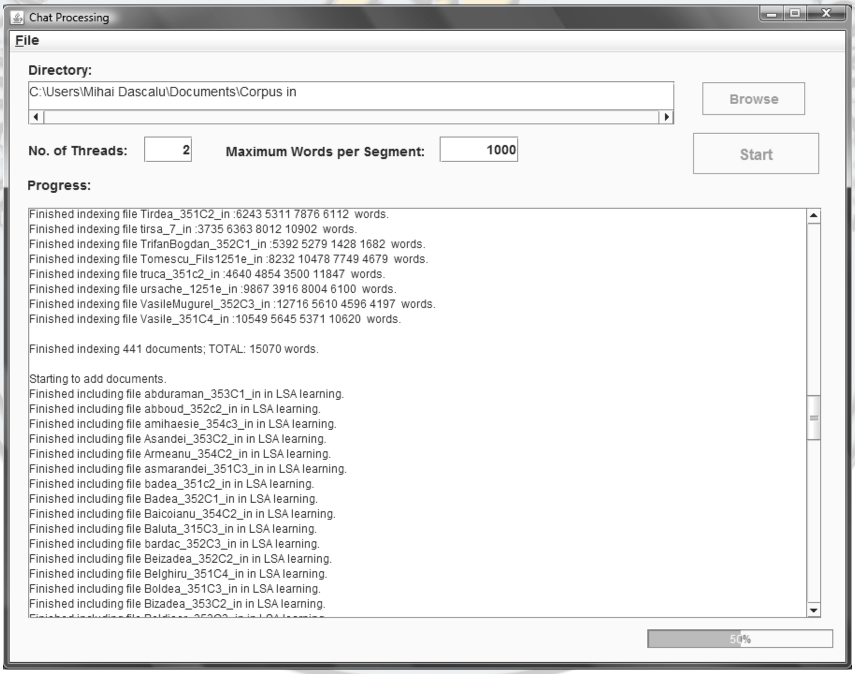

Fig. 8. The LSA learning program interface

3.2. Tf – Idf

A common method for weighting the elements of the term-document matrix is Tf – Idf (term frequency - inverse document frequency [6]) which provides a practical approach for obtaining a 2 part weight for each term taking into consideration all documents:

- Term frequency normalizes the number of appearances of a word in a document;
- Inverse document frequency influences the overall weight by evaluating the appearances of a given word in all documents of the corpus (rare words are given an important bonus, whereas common words receive a lower weight).

The final weight is obtained using the following equation:

$$w_{D,i} = (1 + \ln(tf_{D,i})) \times \ln \frac{N}{n_i}, \quad (5)$$

where $tf_{D,i}$ is the number of occurrences of the term i in document D, N is the total number of documents in the corpus and n_i is the number of documents in which the term i is present.

3.3. The Learning Process

Instead of using regular corpora containing text documents, the designed system uses words from chats and their synonyms (synsets) obtained from WordNet (http://wordnet.princeton.edu), a large English lexical database in which words are grouped into sets on synsets, each expressing a distinct concept – therefore similar to the LSA approach of projecting words, grouping them into concepts and reducing the problem dimension. Synsets are interlinked by means of lexical and conceptual-semantic relations making WordNet a very useful instrument in natural language processing. The use of WordNet is justified by the few and dispersed words in each chat utterance, thus providing the means to increase a word's semantical domain.

The learning process steps are:

1. Word indexing:
- eliminate stop words (very frequent and irrelevant words like "the", "a", "an", "to", etc.) from each utterance;
- apply spellchecking, stemming and again spellchecking for each remaining word;
- enlarge each stemmed word's domain using synsets from WordNet - the relations taken into consideration are synonyms, hypernyms and hyponyms, each with a predefined influence;
- include these words in the list of words taken into consideration;
- in this stage, the total number of documents (sum of number of participants per chat) is computed, making possible the adjustment of the segmentation window size.

2. The effective learning process:
- add each document for all the participants from the corpus;

- once the term-document matrix is populated, apply Tf-Idf and singular value decomposition (SVD) to obtain the final decomposition;
- the dimension of the array is reduced to a dimension *k*.

An important aspect that needs to be taken care of is the value for *k*. This is how LSA smoothes the data, from an initial rank to a more manageable rank, empirically selected in the range of 100 to 300.

4. System evaluation

For adjusting the weights of each factor, machine learning algorithms and an annotated corpus are needed. This allows fine tuning of the evaluation tool and for this purpose another component has been developed.

The "Chat Evaluation" System analyses the performance and correction of ChAMP by comparing the results with those form the golden standard. This is done in parallel using the "Replicated Workers" schema: for each chat, ChAMP evaluation is performed, final grade is converted to a scale of 1 to 10 using a linear distribution and saved in an XML file and, in the end, it is compared to the grade given by an annotator.

Two measurements were evaluated: relative and absolute correctness for each participant's grade. Relative and absolute correctness represent distances between the annotator's grade and the one automatically obtained using ChAMP.

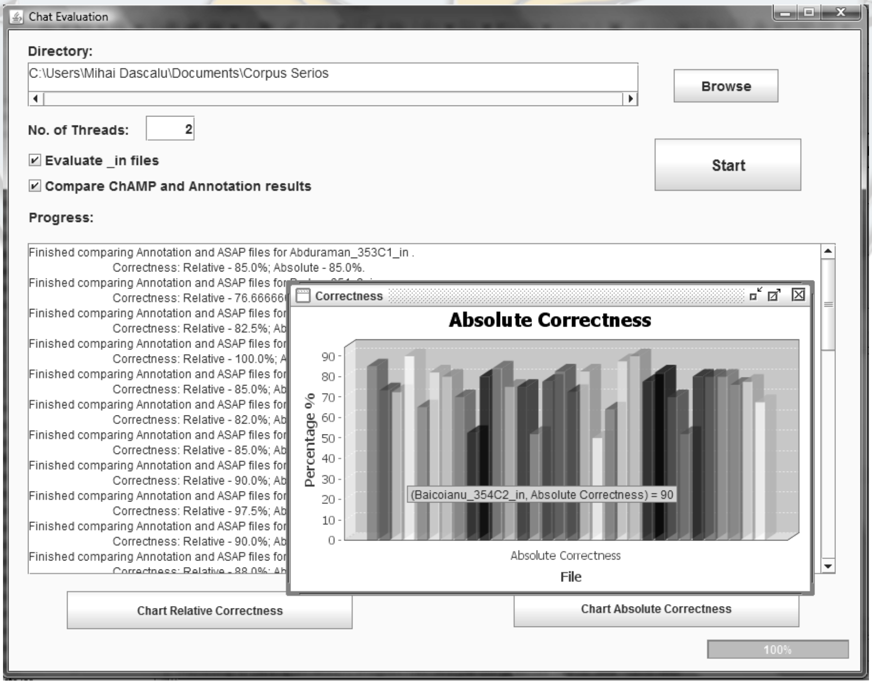

Fig. 9. Main interface for corpus evaluation and overall correctness computation

The average results obtained for the corpus are promising (about 85% relative correctness and 75% absolute correctness) [2].

We strongly believe that with further tuning of the weights, better LSA learning and increased number of social network factors (like betweenness) the results will improve.

Moreover, the subjective factor in manual evaluation is also present and influences the overall correctness.

Conclusions

The first results in using a system conceived from two parts:

- learning from chats using LSA and enlarging the content of each utterance with semantically similar words obtained from WordNet;
- evaluation based both on Social Networks, LSA and Natural Language Processing allow us to conclude that the evaluation of a participant’s overall contribution in a chat environment can be achieved.

In the future, the following improvements are in sight:

- Obtain a larger social network by merging multiple chats – overall evaluation on the entire corpus;
- Semantic segmentation using genetic algorithms;
- Defining patterns, improvements in utterance type determination and speech acts determination to correlate interventions and obtain more specific implicit references;
- The use of reverse indexing to determine the most competent participant overall;
- Profiling each participant from the social networks’ point of view and also with a semantic approach.

REFERENCES

[1] *Mihai Dascălu, Erol-Valeriu Chioaşcă, Ştefan Trăuşan-Matu*, „ASAP – Sistem avansat de evaluare a participanţilor la un chat", Lucrările celei de-a cincea ediţie a Conferinţei Naţionale de Interacţiune Om-Calculator, MatrixRom, 2008

[2] *Mihai Dascălu, Erol-Valeriu Chioaşcă, Ştefan Trăuşan-Matu*, „ASAP – An Advanced System for assessing chat participants", in D. Dochev, M. Pistore, and P. Traverso (Eds.): AIMSA 2008, LNAI 5253, Springer, 2008, pp. 58–68.

[3] *Cormen, T., Leiserson, C., Rivest, R., Stein, C.*, "Introduction to Algorithms", MIT Press 2001

[4] *Holmer, T., Kienle, A. & Wessner, M.* "Explicit Referencing in Learning Chats: Needs and Acceptance", in Nejdl, W., Tochtermann, K., (eds.), "Innovative Approaches for Learning and Knowledge Sharing", First European Conference on Technology Enhanced Learning, EC-TEL 2006, Lecture Notes in Computer Science, 4227, Springer, pp. 170-184.

[5] *Thomas K Landauer, Peter W. Foltz, Darrell Laham*, "An Introduction to Latent Semantic Analysis", Discourse Processes, 25, 259-284, University of Colorado

[6] *Manning, C., Schutze, H.*, "Foundations of Statistical Natural Language Processing, MIT Press: Cambridge Mass, 1999.

[7] *Stahl, G. Group Cognition*, "Computer Support for Building Collaborative Knowledge", MIT Press, 2006.

[8] *Trausan-Matu, S., Chiru, C., Bogdan, R.,* "Identificarea actelor de vorbire în dialogurile purtate pe chat", în Stefan Trăusan-Matu, Costin Pribeanu (Eds.), Interactiune Om-Calculator 2004, Editura Printech, Bucureşti, pp. 206-214.

[9] *Trausan-Matu, S., Stahl, G., Sarmiento, J.* "Supporting Polyphonic Collaborative Learning", E-service Journal, vol. 6, nr. 1, Indiana University Press, pp. 58-74, 2007.

[10] http://www.smashingmagazine.com/2007/06/05/google-pagerank-what-do-we-really-know-about-it/

[11] http://jazzy.sourceforge.net

[12] http://snowball.tartarus.org/

[13] http://tartarus.org/~martin/PorterStemmer/

[14] http://sourceforge.net/project/showfiles.php?group_id=24260

[15] http://prefuse.org/