Using PageRank for Detecting the Attraction between Participants and Topics in a Conversation

Costin Chiru, Traian Rebedea and Adriana Erbaru
University Politehnica of Bucharest, Department of Computer Science and Engineering,
313 Splaiul Independetei, Bucharest, Romania
{costin.chiru, traian.rebedea}@cs.pub.ro, adriana.erbaru@cti.pub.ro

Keywords: CSCL, Natural Language Processing, Participant Assessment, PageRank, Online Conversations.

Abstract: In this paper we present a novel approach that uses the well-known PageRank algorithm for assessing multi-threaded chat conversations. As online conversations can be modelled as directed graphs, we have investigated a method for allowing a real-time analysis of the conversation using PageRank by computing the ranks of the utterances based on the explicit and implicit links available in the discussion. This model has been also extended to offer a method for computing connections between the debated topics and the chat participants and between each of the debated topics in the conversation, called the participant-topic and the topic-topic attraction. The results presented in this paper are promising, but also reflect several important differences between the existent offline analysis tools for chats and the PageRank method.

1 INTRODUCTION

Chat conversations (instant messaging) represent nowadays one of the most popular methods of exchanging ideas online. The easiness in learning how to use chats and the high efficiency in transferring the information, promoted chats as one of the favourite environments for Computer Supported Collaborative Learning (CSCL) tasks requiring online and synchronous textual interactions among participants (Stahl, 2006; Stahl, 2009). Due to this fact, it has been largely adopted in CSCL activities and it has been enhanced with functionalities specific to these tasks such as the explicit referencing mechanism and the whiteboard facility present in ConcertChat (Muhlpfordt and Wessner, 2005). Still, in spite of its popularity and of the huge quantity of data that is exchanged through chats, there are very few application aimed at analyzing this type of content (Chiru et. al, 2011; Rebedea et. al, 2011). More than that, the existing applications are built starting from a semantic analysis (Chiru et. al, 2011; Rebedea et. al, 2011) but the analysis takes far too much time to be used as a real-time process and can only be applied offline, at the end of the conversation. Therefore, we have been searching for a different method to analyze these conversations faster and got influenced by the algorithms which are used by search engines that have to analyze huge quantities of data in a very short time. Thus, we reached the conclusion that if the PageRank algorithm (Page et. al, 1998) could be adapted for chats, this method could be applied online (displaying the results of the processing as the conversation unfolds) and interactive (to signal what threads should be debated more and involving people who contributed less on specific threads), this way improving the learning process and enhancing the participants' innovation.

To achieve this, we started from PolyCAFe (Rebedea et. al, 2011), a system that is using innovative methods for analysing CSCL chat transcripts, helping both computer-assisted learning and the tutors in evaluating the discussions. This system analyzes chat logs using Natural Language Processing (NLP), Latent Semantic Analysis (LSA) and Social Network Analysis (SNA) techniques in order to identify the most important utterances from the conversation (in terms of their content and of the participants' involvement in the discussion).

Therefore, this paper presents an extension of PolyCAFe's functionality, trying to enhance it with the ability to analyze CSCL chat sessions in real-time. The first step of our analysis consists of detecting the important utterances from the conversation using the PageRank algorithm. Once these utterances are identified, we use the PageRank algorithm for detecting what is the attraction of each participant towards the debated topics and, at the same time, what is the probability of a topic to follow another topic within the conversation.

Most of the existing approaches for analyzing text using social network analysis (SNA) tools are oriented towards systems that own explicit referencing tools, such as forums or blogs. The reason behind this orientation is the ease in constructing the participant social network based on the order in which the messages are sent and on the recipient of the message. Still, there are a few systems that intended to apply SNA tools to chat conversations. One such tool was built by Sundararajan (2010) for analyzing the content published by the participants to 8 different courses in order to observe how the respect and influence earned by each participant influences their efforts to "collaborate, learn new and conceptual knowledge" and their satisfaction regarding the courses outcome. Unfortunately, the author does not mention whether this analysis is done manually or automatically. Moreover, a regular SNA method is used for evaluating the participants from the perspective of their centrality, betweenness, in-degree, out-degree, etc. in the network, which represent only quantitative data. On the other hand, we are rather interested in what the participants communicate (what are the topics they know or they are interested on) and in the interaction patterns between different concepts that are debated in the conversation, which is part of a qualitative evaluation of the participants, topics and the conversation as a whole.

A more similar approach was undertaken by Tuulos and Tirri (2004). The authors present a semi-supervised system that uses a combination of topic modelling and SNA to improve the information retrieval from chat conversations. For their analysis, they have used conversations taken from SearchIRC.com which allowed them to use simple heuristics in order to identify to whom each utterance is addressed (and therefore to build the social network). For this participant network the in-degree, out-degree and PageRank of each participant are determined. After that, the authors use some existing conversations to detect the probabilities of words to appear in conversations about different topics, so that when they analyze new conversations to be able to use these probabilities. Finally, they evaluate the use of each of the SNA technique in improving the information retrieval, considering as baseline the results provided by the topic modelling. Still, this approach gives them two advantages: first of all they know both how many and what topics should be present in the conversation (therefore knowing what represents off-topic and being able to discard that part); secondly, they have chosen the topics from different topics (Bible, C++, Philosophy, Physics, Politics, Win2000) thus simplifying the task of identifying to what topic a given concept corresponds. In our approach neither of these facts can be exploited: since our system does not have a learning phase, it gives the possibility to analyze texts debating about any topics, without being limited to the ones that were learnt (thus providing generality in use). At the same time, it can be used to distinguish between concepts that are from the same or similar conceptual area. The examples presented in this paper contain concepts from a single domain (Human Computer Interaction) especially to prove that the approach works even at this level, without requiring that different topics to be debated in the same conversation.

The paper continues with a short overview of the PageRank algorithm. Then, we present the application that has been developed and several results that have been obtained by employing the PageRank method adapted for CSCL chats. The paper ends with an analysis of these results and with our conclusions regarding the improvement of the results' quality.

2 OVERVIEW OF THE PAGERANK ALGORITHM

Because previous researches have modelled an online conversation as a graph with implicit and explicit links between utterances (Rebedea et. al, 2011), we have started to consider that the PageRank algorithm (Page et. al, 1998) may be a candidate for the conversation graph analysis. PageRank is an algorithm that was initially designed for the analysis of a set of web pages in order to extract the relative importance of each page from the considered set of web pages (Page et. al, 1998). The algorithm expresses the probability that a web surfer will be able to "find" the considered page within a limited number of steps (clicking on the links from one page to another). It is a customization of a "random walk" in a graph, which in turn is modelled as a Markov chain in which the states are pages, and the transitions, which are all equally probable, are the links between pages.

The formal definition given in the initial paper describing PageRank (Page et. al, 1998) was: if *u* is a web page; *Fu (forward links)*, the pages referred by *u; $Nu = |Fu|$* the number of forward links; *Bu (backward links)* the ones that refer *u*, *c* a normalization constant and E(*u*) a source of rank to make up for the rank sinks (such as cycles) with no out-edges, than the value of the rank R(*u*) can be computed using:

$$R(u) = c * \sum_{v \in B_u} \frac{R(v)}{N_v} + c * E(u) \tag{1}$$

In order to compute the vector R(*u*), one starts from the square matrix (we'll call it A) having the web pages on the rows and columns and $A[u, v] = 1/Nu$ if there is a link from page u to page v or 0 otherwise. If R is the a vector of scores over the web pages, then we can write $R = c(AR + E)$, which can be re-written as $R = c(A + E \times 1)R$ because the values of *PR* are normalized and therefore $||R||_1 = 1$. That means that R is the eigenvector of $A + E \times 1$ and the method should also try to maximize the value of c (Page et. al, 1998). Subsequent research showed that the optimal value for c should be 0.85 (Brin and Page, 1998).

The value of R can be obtained in an iterative manner, starting from a vector of values over the web pages (*S*) that can have any values (could be the vector *E(u)*), using the following iterative algorithm (Page et. al, 1998):

```
$R_0 \leftarrow S$
$loop$:
    $R_{i+1} \leftarrow A_i$
        $d \leftarrow ||R_i||_1 - ||R_{i+1}||_1$          (2)
    $R_{i+1} \leftarrow R_{i+1} + dE$
        $\delta \leftarrow ||R_{i+1} - R_i||_1$
$while\ \delta > \epsilon$
```

, where d is a factor for increasing the convergence rate and for maintaining $||R||_1 = 1$, while $\epsilon = 10^{-3}$.

A web page will have a high PageRank if the sum of the webpages' PageRank that refers it is large. This property covers two possible cases: when a page has many other pages referring it, or when it is referred by pages with high PageRank.

The PageRank algorithm has proved to be suitable not only for Google's rank of web pages, but also for other tasks in various domains: replacing the ISI factor with a new formula based on the PageRank Algorithm (Bollen, Rodriguez and Van de Sompel, 2006), ranking academic doctoral programs based on their records of placing their graduates in faculty positions (Schmidt and Chingos, 2007), predicting how many people (pedestrians or vehicles) come to the individual spaces or streets (Jiang, 2006), performing Word Sense Disambiguation (Navigli and Lapata, 2010), etc. Thus, we hoped that it could also work for chat analysis especially as a conversation can be seen as a graph of links between utterances where discourse flows in a similar manner to the importance of the web pages.

In order to use this approach, we considered that each utterance from the chat conversation represents a different document and in order to simulate the forward and backward links, we used explicit and implicit links (details will be provided in the next section). Thus, we managed to develop a method for very fast identification of the important utterances from a chat, of the major threads of discussion, and of the participants' attraction towards these threads.

3 PAGERANK FOR AUTOMATIC ASSESSMENT OF CSCL CHATS

As we have already mentioned, our application starts from PolyCAFe project and uses some of its features:

- Detection of the areas of high collaborative discourse from a chat;
- Evaluation of the collaboration of each participant in the discussion (based on multiple criteria);
- Graphical representation of the results.

3.1 Pagerank for Conversation Analysis

We consider that PageRank is appropriate for chat evaluation as this problem is very similar to the original problem for which it was initially designed. The sparsity in the chat is (in our opinion) similar to the sparsity of relevant content from the web. Therefore, we consider that the probability of a person to "land" on a specific page from the web is similar to the probability of a participant to reply to a given utterance, while the links between different pages are well simulated by the semantic connections represented by the repetitions of the same word and by the explicit "reply to" links. Practically, this probability of a participant to reply to a given utterance can be considered the "rank of an utterance" (and it highlights the importance of that specific utterance in the conversation).

The first step in applying the PageRank Algorithm was the identification of the links that exist between chat utterances. Since the pre-processing part of our application was borrowed from PolyCAFe, we also kept the input format of the chats, which allowed the existence of explicit links (references provided by the chat participants to specify to which previous utterance their answer is addressed).

Besides the explicit links, one can also encounter the situation when two or more utterances contain concepts that are strongly related and therefore their authors consider that there is not necessary to provide an explicit link. We considered that this situation is a special case of connection (an implicit link) and tried to identify it in order to augment the number of explicit connections (that was insufficient for our purposes). Therefore, we considered that words repetitions (Chiru et. al, 2011) are example of such links. If a term appears in an utterance, all lines that follow and contain that term are considered implicit links to the initial utterance. Given the nature of the algorithm, the two types of links that we consider (explicit and implicit) have equal weight.

Once we detect all these links, we build utterance chains (which can be interpreted as discussion threads since they debate the same concepts) starting from these links using the DFS (Depth First Search) Algorithm, thus finding all the existing separate chains. They are needed for determining the attraction between two different threads (topics).

The steps that should be followed in order to determine the threads are:

1. Identify all the utterances that are not referenced (neither by explicit nor by implicit links) – these utterances are probably off-topics and therefore they are ignored;
2. All the remaining utterances are considered to be roots for the DFS Algorithm;
3. From each of these utterances (considered in the order they appear in chat) we start a function (implementing DFS) to detect the threads that can be built starting from that utterance;
4. Each function will return a thread of utterances.

The next step is to create the transition matrix corresponding to the chat utterances by considering the links identified between them. The explicit and implicit links between two utterances will provide a value of 1 in the matrix, while the remaining elements are set to 0 (meaning that there is no connection between the corresponding two utterances). Once this matrix is built, it needs to be normalized with respect to the sum of the elements from each column.

Finally, the values of the PageRank algorithm for the given matrix are obtained using the power method implementation provided by the JAMA library - A Java Matrix Package (Hicklin et. al, n.d.). The operations made for the detection of the eigenvalues and the eigenvectors are:

1. Apply the *eig* method, which decomposes the matrix in two other matrices: a matrix *D* containing the eigenvalues and a matrix *V* containing the eigenvectors;
2. The maximum (dominant) eigenvalue from the diagonal matrix *D* is determined and its index is stored;
3. The dominant vector is the column from the matrix *V* having the index identified in the previous step;
4. The values from this vector are normalized with respect to the sum of its elements;
5. The final values (the PageRank) are the values obtained for the normalized eigenvector *vd* (*utterance[i].rank = vd[i]*).

Once these values are determined, one can evaluate the participant-topic attraction and the topic-topic attraction as described in the following sections.

3.2 Participant-topic Attraction

The *participant-topic attraction* defines the participants' drive to get involved in the discussion of a given concept therefore proving its interest or knowledge related to that concept. To determine this factor, we have used the values of the participant's utterances containing the words that define the considered topic.

If *p* represents a participant and *t* a topic, then the attraction between p and t is given by the following formula:

$$a(p,t) = \sum_{\substack{up \in Utt \\ up \in p}} rank(up) * freq(t, up) \qquad (3)$$

In order to highlight this method, we provide an example that proves how the above formula works. For this, we have made the simplifying assumption that all the utterances belong to the same participant.

Utt:	**< debated topics >**	**utt value**
u1:	t t y x	0.5 = rank(u1)
u2:	t y z x	0.3 = rank(u2)
u3:	t t t z	0.2 = rank(u3)

Using formula (1), the following results will be returned (2):

$$a(p,t) = 2 * rank(u1) + rank(u2) \qquad (4)$$

$$+3 * rank(u3)$$
$$a(p,x) = rank(u1) + rank(u2)$$
$$a(p,y) = rank(u1) + rank(u2)$$
$$a(p,z) = rank(u2) + rank(u3)$$

In the end, all these values are normalized.

3.3 Topic-Topic Attraction

The *topic-topic attraction* defines the probability of having a specific topic following another topic in the flow of a conversation. To determine it, we use the utterance chains taking into account both the frequency of each topic and the case when they co-occur in the same utterance (topics are very closely related) or occur separately (more loosely).

Therefore, we extract the threads corresponding to the two topics and build a matrix for each chain. This matrix reflects the debating of those topics in the utterances and their corresponding values within that chain.

The relationship between the topics and the utterances is reflected by the matrix that is built as follows:

1. We build the *chain – topic matrix* (*ct*), $ct[i][j]$ represents the value of the j topic in the i utterance where
   a. $ct[i][j] = 0$ if the i topic is not debated in the utterance j;
   b. $ct[i][j] = \frac{rank(utt\_j)}{\#topics\ in\ utt\_j}$ if the i-th topic is debated in the utterance j.
2. After filling in the matrix, we apply formula (5) for each topic t_i and t_j.

$$\frac{\sum_{u\in utt\_chain} ct[u][t_i] + \sum_{u\in utt\_chain} ct[u][t_j]}{\sum_{\substack{u \in utt\_chain \\ t \in t_i | t_j}} ct[u][t]} \qquad (5)$$

Below we present an example of matrix (6) for determining the *topic – topic attraction* for the following chain: u5 → u4 → u3 → u2 → u1

utt:	**<debated topics >**		
u1	t1	t2	t3
u2 :	t1		t5
u3 :	t2		t4
u4 :	t1		t3
u5 :	t1		t2

Then, the attraction between topic t1 and t2 topic is given by (5) by applying the formula (3), with the matrix from (4), obtaining formula (7).

3.4 User Interface

The user interface allows the input file selection, and afterwards the content of this file is analyzed and the results are displayed in tabular form (see Fig. 1).

The left part of the GUI presents the values for participant – topic attraction for the selected participant, while the right part gives the values of the topic – topic attraction for the selected topic.

Besides the values obtained for the participant – topic attraction and the topic-topic attraction, the application also outputs the most important utterances from the chats computed using their PageRank.

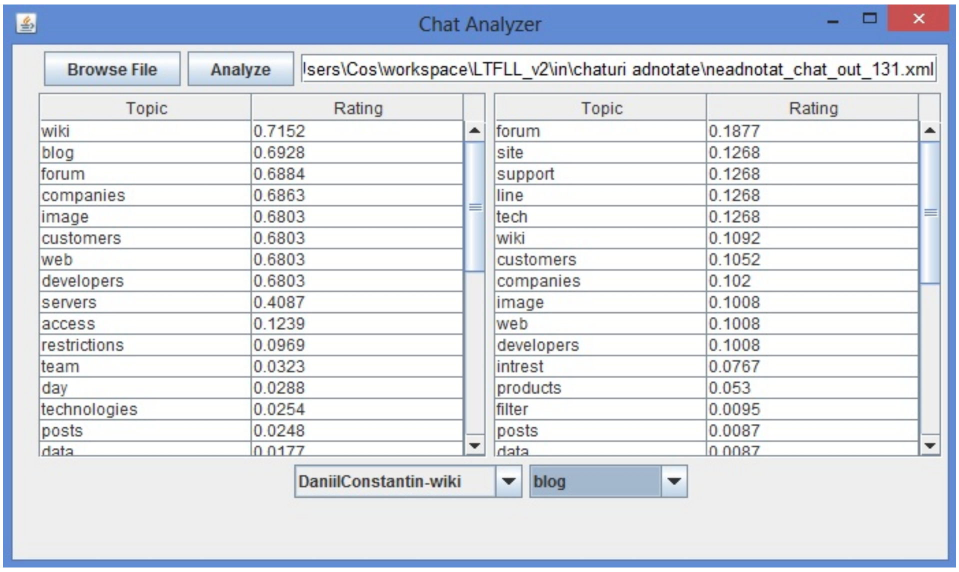
The results proved to be much stricter comparing to the results obtained using PolyCafe system or provided by the human reviewers (Gold-Standard). This is due to the fact that only very few utterances have a PageRank greater than 0.

In order to provide an example, we present a part of the utterances evaluated as being important by PageRank algorithm. We will use the same chat for which we presented the examples from the participant-topic and topic-topic attraction examples. The automatic analysis performed with PolyCAFe has identified 132 important utterances (out of 430) as important. From these, the PageRank algorithm also identified 17 utterances as (see Table 1). The values for the ranks computed by PageRank may seem pretty low, but this is what usually happens when applying the algorithm on any graph.

Besides these 17 utterances that were considered important by both PolyCAFe and the PageRank method presented in this paper, the latter has identified another 28 turns that were not considered important by the former. In order to account for these utterances, we analysed PolyCAFe's results in order to discover a possible explanation. At a careful analysis, we observed that the 28 extra utterances were marked by PolyCAFe as being continuations of other utterances. Therefore, it is possible that PolyCAFe did not consider these utterances to be

$$ct = \begin{pmatrix} rank(u1)/3 & rank(u1)/3 & rank(u1)/3 & 0 & 0 \\ rank(u2)/2 & 0 & 0 & 0 & rank(u2)/2 \\ 0 & rank(u3)/2 & 0 & rank(u3)/2 & 0 \\ rank(u4)/2 & 0 & rank(u4)/2 & 0 & 0 \\ rank(u5)/2 & rank(u5)/2 & 0 & 0 & 0 \end{pmatrix} \qquad (6)$$

$$a(t1,t2) = \frac{rank(u1) * 2/3 + rank(u2)/2 + rank(u4)/2 + rank(u5)}{rank(u2)/2 + rank(u3)/2} \qquad (7)$$



!sers\Cos\workspace\LTFLL_v2\in\chaturi adnotate\neadnotat_chat_out_131.xml

Topic	Rating
wiki	0.7152
blog	0.6928
forum	0.6884
companies	0.6863
image	0.6803
customers	0.6803
web	0.6803
developers	0.6803
servers	0.4087
access	0.1239
restrictions	0.0969
team	0.0323
day	0.0288
technologies	0.0254
posts	0.0248
data	0.0177

Topic	Rating
forum	0.1877
site	0.1268
support	0.1268
line	0.1268
tech	0.1268
wiki	0.1092
customers	0.1052
companies	0.102
image	0.1008
web	0.1008
developers	0.1008
intrest	0.0767
products	0.053
filter	0.0095
posts	0.0087
data	0.0087

DaniilConstantin-wiki

blog

Figure 1: Application Graphical User Interface.

Table 1: The utterances identified by our algorithm that receive a grade higher than 8 by PolyCAFe.

Utt. No	PageRank score	PolyCAFe score	Utterance Content
169	0.002	10.07	yes, they have wikis that are publicly available, with public information, for the everyday user that takes an intrest in that company's products
167	0.002	10.01	all major companies have wikis for their technologies. most people like to search wikis cause they provide accurate and easy to access information. Also, that way our database servers won't be so used
404	0.202	10.01	Indeed. Our companies image will grow if we have a forum, a blog, a wiki and a cool web-site that customers or developers can use
310	0.004	9.93	the only problem that still remains is that we need someone to check wiki articles, blog and forums posts so that classified data does not accidentally reach a "public" area
348	0.004	9.28	A svn is an open-source revision control system. Users can work on a version of the application code and commit it. If two users are working on the same thing when they commit a merge is made with the 2 versions
308	0.001	9.03	we can use a person or a team of people to handle the wiki posts, forums posts, wave documents and all the other important stuff
331	0.002	8.93	I mean everyone of our employees knows how to use a wiki, forum blog and chat, and google wave has a extremely friendly interface
314	0.003	8.85	but how can you use a filter for a forum?
399	0.009	8.79	well i think chat is important for our employees, it helps them talk and colaborate, spare time by not meeting in conferences that much, and be on track with all theit colleagues are doing
312	0.002	8.78	We can use filters for that firewalls. That can save some money
388	0.047	8.45	not necesarly computers,you can change acounts
333	0.001	8.4	evrybody can use a chat , forum , bog or something like that
423	0.015	8.32	good night everyone, and thank you for your collaboration:)
341	0.004	8.21	We can also use a SVN for our code. What do you think of this?
376	0.042	8.12	this could make them loose time...
373	0.029	8.08	they will use another machine. the restrictions will be only for certain computers
387	0.025	8.07	for every function you don't remember in a programming language, you will have to move to other computer to find out... but it's ok ... it wouldn't be a big problem i guess

important, since the same ideas were present in the previous utterances, but PageRank, through its nature, favours this kind of utterances since being identified as continuations it means that they have links from other utterances that were considered important in the past and therefore they receive a part of these utterances' rank.

For a better evaluation of our method, we asked 30 students from the Human-Computer Interaction class to annotate 3 different chat conversations with the most important utterances. Thus each of the three chat conversations was evaluated by 10 different students. We computed the inter-rater agreement using Fleiss' Kappa for m raters and we have obtained the values for Kappa 0.133 for the first chat, 0.142 for the second and 0.177 for the third one, while the p-values were always 0.000. These results show how difficult this task is even for humans. When we computed the results obtained by our method with the gold standard results provided by the annotators, we obtained the values of kappa 0.085 for the first chat, 0.0882 for the second and 0.0894 for the last one. These results are below those of the raters, but we have remarked that if we discard the last 15% of the utterances in all chat conversations, where the PageRank accumulated too much, the results are much better: 0.131 for the first chat, 0.128 for the second and 0.173 for the last chat conversation. These results are closer to the inter-rater agreement and highlight that we should add a decaying factor for utterances that are closer to the end of the discussion (as they have fewer out-going links and thus the rank tends to accumulate in them).

4 INTERPRETATION OF RESULTS

There are several important observations that can be drawn up based on the results that we have obtained and analyzed. First of all, the computed ranks for the utterances are rarely different from 0, this fact being generated by several reasons:

- Most of the chats contain very few explicit links (they seem to be ignored quite often by the chat participants) – we have observed a direct dependence between the number of explicit links from the chats and the number of utterances having non-zero values after applying the PageRank algorithm.
- The PageRank algorithm determines the utterances' ranks as a random walk in the graph of utterances. The significance of these values is the probability to get to a certain utterance after a number of steps that goes to infinity. Therefore, once the algorithm gets to a (relatively small) set of utterances (lines / columns from the transition matrix), it will be very difficult to get out of that set (in the context of random walk) and so the remaining values will tend to be 0.
- The PageRank Algorithm is designed for the web, where a lot of links exist between different resources (therefore creating large chains of links, most of the times having a lot of cycles), while in the chats the utterance chains are usually short and rarely having such cycles. Besides, the topics repetitions might not be synchronized with the explicit references, therefore not leading to cycles.
- In the current version of the system, we proposed equal importance for the explicit and implicit links, which can lead to determining a value too high / low for some utterances, depending on the number of words used in that utterance and in the one to which it is linked.

Secondly, most utterances having values greater than 0 are positioned at the end of the discussion. This happens due to lack of explicit links, and therefore those utterances accumulate a very high score due to topics repetition, which propagates through the chat from the beginning until the end. A solution is needed to link these utterances to other ones in the chat.

Finally, there are some utterances considered significant by the PageRank algorithm, but not by other algorithms or by human evaluators. The reason is the same as for the previous observation: some utterances (that are not very significant in terms of conversation) may receive high values because of the rank accumulation over time from other utterances that contain the same topics.

There are a couple of solutions that could be tried in order to alleviate the presented problems. A first solution might be the detection of dialog acts and adjacency pairs, since the main problem of the proposed method is the lack of explicit links. This way, one can detect the dialog acts that are present in the chat (question - answer, agreement – disagreement, greetings and so on) very quickly and to use these links as additional explicit links. Another possibility is to use LSA or lexical chains to find out more semantic connections between utterances.

Another solution to avoid reaching too many zeroes for the computed ranks is to use the Iterative Method instead of the Power Method for computing the PageRank values. This way, one is not constrained to apply the algorithm until convergence (after an infinite number of steps), but can stop after a limited number of steps, so that fewer utterances reach a zero-value influence.

Finally, in order to be able to discriminate between the importance of explicit and implicit links, one can use a generalized algorithm based on Markov chains having different values for different link types (explicit or implicit links).

5 CONCLUSIONS

To sum up, our current adaptation of the PageRank algorithm for online conversations (using only the explicit and implicit links given by the topics repetitions) is not powerful enough to provide results that have the desired accuracy compared with other solutions that analyse the discussions offline. The main explanation is that there are not enough explicit links added by the participants during the discussion and using only repetitions for detecting implicit links does not build a graph that is dense enough. However, the method is much faster and it can be used online and in real-time for the dynamic evaluation of multi-threaded discussions involving multiple participants.

Moreover, in our opinion the assumptions made in this paper are novel for the analysis of online discussions and they have not been used to assess the importance/rank of an utterance in an online discussion although PageRank follows from previous work in citation analysis (where the links between papers are made explicit by authors). The preliminary results also support the use of PageRank to compute the most important utterances in a multi-party online conversation, but several improvements of this method need to be investigated in order to achieve similar results to the current state of the art methods that also employ linguistic analysis.

ACKNOWLEDGEMENTS

This research was supported by project No.264207, ERRIC-Empowering Romanian Research on Intelligent Information Technologies/FP7-REGPOT-2010-1.

REFERENCES

Bollen, J., Rodriguez, M. A. and Van de Sompel, H. 2006. Journal Status. In: *Scientometrics 69 (3)*, pp. 669-687.

Brin, S.; Page, L. 1998. The anatomy of a large-scale hypertextual Web search engine. In: *Computer Networks and ISDN Systems 30*: 107–117.

Chiru, C., Cojocaru, V. Trausan-Matu, S., Rebedea, T. and Mihaila, D. 2011. Repetition and Rhythmicity Based Assessment for Chat Conversation. In: *ISMIS 2011, LNCS 6804*, Springer, pp 513-523.

Hicklin, J., Moler, C., Webb, P., Boisvert, R., Miller, B., Pozo, R., Remington, K. Jama: a Java matrix package. (http://math.nist.gov/javanumerics/jama/ - accessed 16/08/2012).

Jiang, B. 2006. Ranking spaces for predicting human movement in an urban environment. In: *International Journal of Geographical Information Science 23 (7)*, pp. 823–837.

Muhlpfordt, M. and Wessner, M. 2005. Explicit referencing in chat supports collaborative learning. Paper presented at the *Proceedings of CSCL 2005*.

Navigli, R. and Lapata, M. 2010. An Experimental Study of Graph Connectivity for Unsupervised Word Sense Disambiguation. In: *IEEE TPAMI, 32(4),* IEEE Press, pp. 678–692.

Page, L., Brin, S., Motwani, R., Winograd, T. 1998. *The PageRank Citation Ranking: Bringing Order to the Web*, Technical Report. Stanford InfoLab.

Rebedea, T., Dascălu, M., Trausan-Matu, Armitt, G., and Chiru, C. 2011. Automatic Assessment of Collaborative Chat Conversations with PolyCAFe. In: *Proceedings of ECTEL 2011, LNCS 6964*, Springer, pp. 299-312.

Schmidt, B. M. and Chingos, M. M. 2007. Ranking Doctoral Programs by Placement: A New Method. In: *PS: Political Science and Politics 40*, pp. 523–529.

Stahl, G. 2006. *Group cognition. Computer support for building collaborative knowledge*. Cambridge: MIT Press.

Stahl, G. 2009. *Studying Virtual Math Teams*. New York: Springer.

Tuulos, V. H. and Tirri, H., 2004. Combining topic models and social networks for chat data mining. In: *Proceedings of WI'04,* pp. 206–213.

Sundararajan, B., 2010. Emergence of the Most Knowledgeable Other (MKO): Social Network Analysis of Chat and Bulletin Board Conversations in a CSCL System. *Electronic Journal of E-Learning, 8(2),* pp. 191-207.