ReaderBench Talk: Online Conversations Empowered by Advanced NLP Techniques

Andrei Mardale, Gabriel Gutu-Robu, Mihai Dascalu, Stefan Trausan-Matu
University Politehnica of Bucharest
313 Splaiul Independentei, Bucharest, Romania
andrei.mardale@cti.pub.ro, {gabriel.gutu, mihai.dascalu, stefan.trausan}@cs.pub.ro

ABSTRACT
With the increased need of communication, Computer-Supported Collaborative Learning has gained a broader attention within the last years. Thus, the necessity of using chat applications emerged; however, most solutions lack the integration of Natural Language Processing (NLP) services in order to provide support and guidance, while discussing. Our novel application *Talk* is integrated within the *ReaderBench* framework and is defined by three main features: asynchronous messages specific to chats, automated bots for enhancing participants' involvement and interactions, as well as the usage of advanced NLP functionalities for a posteriori evaluations. *ReaderBench Talk* communicates with the framework through a dedicated endpoint built within the *ReaderBench* API. As preliminary validation, two assignments were given to students from our faculty. Overall, students were able to solve the tasks without encountering critical bugs and they also provided valuable feedback. Their feedback revealed the usefulness of the application, its ease of use, the persistency of data, and the quality of the NLP services. The feedback also helped identify issues, which were mainly related to the user interface and to the communication with the server.

Author Keywords
Chat Conversations; Computer-Supported Collaborative Learning; ReaderBench framework; Natural Language Processing; Chat Bots.

ACM Classification Keywords
I.2.1. ARTIFICIAL INTELLIGENCE: Applications and Expert Systems.

INTRODUCTION
Computer-Supported Collaborative Learning (CSCL) technologies gained attention within the last years for both scientific researchers and for regular users, mainly motivated by the underlying social web and the knowledge building processes [1]. CSCL scenarios frequently use chats, which work similar to instant messaging applications, but can integrate advanced facilities. CSCL is aimed at a learning tasks and regularly involves more than two students. As specific traits, the resulting conversation may include the intertwining of complex discussion threads in a polyphonic manner [2]. The integration of explicit referencing mechanisms helps follow the threads, while mechanisms of automated detection of implicit links using Natural Language Processing (NLP) techniques [3] can automatically create references between utterances when manual annotations are forgotten or omitted [4].

Several experiments were previously performed by our research group by considering a collection of chat conversations [5]. In order to extend the corpus of conversations, the idea of gathering new conversations for future experiments was considered beneficial. The need to develop a new, dedicated chat application emerged, taking into account that the previously used application is tedious to install and use as it requires obsolete technologies. With this aim, our application was designed to easily enable the integration of NLP facilities, to help increase users' interactivity, and to facilitate referencing between contributions.

This paper is structured as follows. The next section describes related work and it is focused on ConcertChat [6], the *ReaderBench* framework [7, 8]. Afterwards follows the presentation of the chat interface, together with its main windows and functionalities, including the automated bots. Feedback gathered from part of the students' that interacted with the application is then provided followed by conclusions.

RELATED WORK
Taking into account that our first experiments on chat conversation date more than 10 years ago [9], the need of gathering a new collection of conversations motivated us into researching existent solutions. After comparing a few of them, we decided to build our own application, which is presented in the next section.

Existing applications
While desktop-based applications are less and less used nowadays, web-based applications gained a lot of popularity within the last years. Out of existing desktop applications, *ConcertChat* was used in previous experiments conducted bu our team. Web apps include popular services such as Skype, Facebook Messenger, Google Hangouts or Slack.

ConcertChat
ConcertChat (https://sourceforge.net/p/concertchat/) [10] is a software application built in Java, aimed at providing CSCL facilities for chat conversations. It is particularly powerful for collaboration purposes, as it offers two panels for content sharing which suppor both text and graphical

elements. Among the most important features, it is worth mentioning the referencing system. References are messages explicitly annotated by participants as replies to prior utterances. The key aspect of ConcertChat is the fact that responses can be posted both to text utterances and to elements drawn on the whiteboard. Due to the dynamic character of chat environments, chaining ideas can be problematic when multiple participants actively discuss; ConcertChat overcomes this impediment with the aid of explicit referencing. Thus, linking a text message to a piece of drawing enriches communication expressivity and reinforces the outlined ideas. Another benefit of the reply-based communication tools is the ability to decrease confusion among participants, as the flow of discussions is enhanced. By using replies, text coherence is improved and discourse becomes more cursive [6].

ConcertChat offers persistent storage of conversations; thus, people who join a room later on can have access to the entire conversation and can easily get involved. Another advantage of the persistency is the fact that scientific analyses can be made a-posteriorly. Both texts and the drawn elements are kept in the history, thus allowing users to easily revert to a discussion to a previous state.

ConcertChat was used for creating of the corpus of conversations used in our previous experiments [11]. As the software application was not updated since 2007 and is based on currently obsolete technologies, the stringent need of building a dedicated application emerged.

Web-based application

One of the main issues of available online applications is the lack of the facility to export conversations for follow-up analyses. Moreover, most applications (e.g., Slack) limit the number of messages that can be retrieved for free usage. Moreover, the majority of such services are also based on central servers that store the entire conversation (sometimes un-encrypted), thus exposing sensitive information. The deletion of conversations is also sometimes a tedious task. Needless to say, there are no NLP facilities integrated within existing online applications.

The need for a dedicated chat application

Our decision was to create a new user-friendly web application, as a dedicated endpoint for analyzing chat conversation was already implemented within the *ReaderBench* Application Programming Interface (API) [12]. The increasing spread of online services in the detriment of desktop applications also motivated the decision to create it as a web interface. Such applications usually work without the need to install additional software, which makes them easier to be used, while increasing the availability of server-side hardware resources for advanced processing. Thus, we are able to use our API in order to render collaboration results and sociograms.

The *ReaderBench* CSCL endpoint

ReaderBench is an advanced multi-lingual Natural Language Processing framework [12]. The assessment of chat conversations is one of the provided functionalities within the Application Programming Interface (API) and it is used for the development of the chat application. A dedicated endpoint is publicly available through POST requests sent to the *ReaderBench API* at http://readerbench.com/api/cscl-processing. This service performs multiple NLP analyses, out of which we highlight:

- Keywords extraction;
- The assessment of users' participation throughout the conversation;
- The evaluation of participants' interactions;
- The assessment of collaboration in regard to social knowledge building, based on Cohesion Network Analysis [13];
- Specific indices, denoting characteristics such as participation and collaboration for each participant.

The CSCL processing service requires the following input parameters to perform a request:

- The chat conversation to be processed, provided in a specific XML format, compatible with ConcertChat;
- The language of the conversations (English and French are currently supported);
- The pre-trained corpus to be used for the integrated semantic models. A corpus consisting of documents comprised of both general and specific terms is recommended; TASA (http://lsa.colorado.edu/ spaces.html) is a good example for current experiments;
- A similarity threshold for calculating semantic similarity scores between concepts; only the scores exceeding the threshold will be retrieved;
- Options for advanced computations such as the enabling of the part-of-speech tagger, which is aimed at detecting words' parts of speech, or the dialogism mechanism, which allows the detection of semantic chains.

The output data provided as response by our CSCL processing service include:

- Participants' interactions,;
- The evolution of participants' contributions to the overall conversation;
- Knowledge brought by each participant to the conversation;
- The manner in which users' points of view (i.e., voices from dialogism) intertwine throughout the conversation;
- The most relevant concepts of the conversation, together with the semantic relatedness links

between them, which may be used for creating a concept map;

- Indices denoting CSCL features specific per user, such as their number of contributions, their degree of inter-animation, the overall score of their contributions or their degree in terms of posted and received messages.

The participants' evolution and interaction graphs, together with the map of concepts have been used within the chat application to visually express the main characteristics of a chat conversation. Information with regards to participants' collaboration and participation effects have been associated also with each utterance in order to highlight intense collaboration sections within the conversation. Our aim is to allow users to easily analyze chats with the help of specific visualizations presented in detail in the following section.

THE CHAT SYSTEM

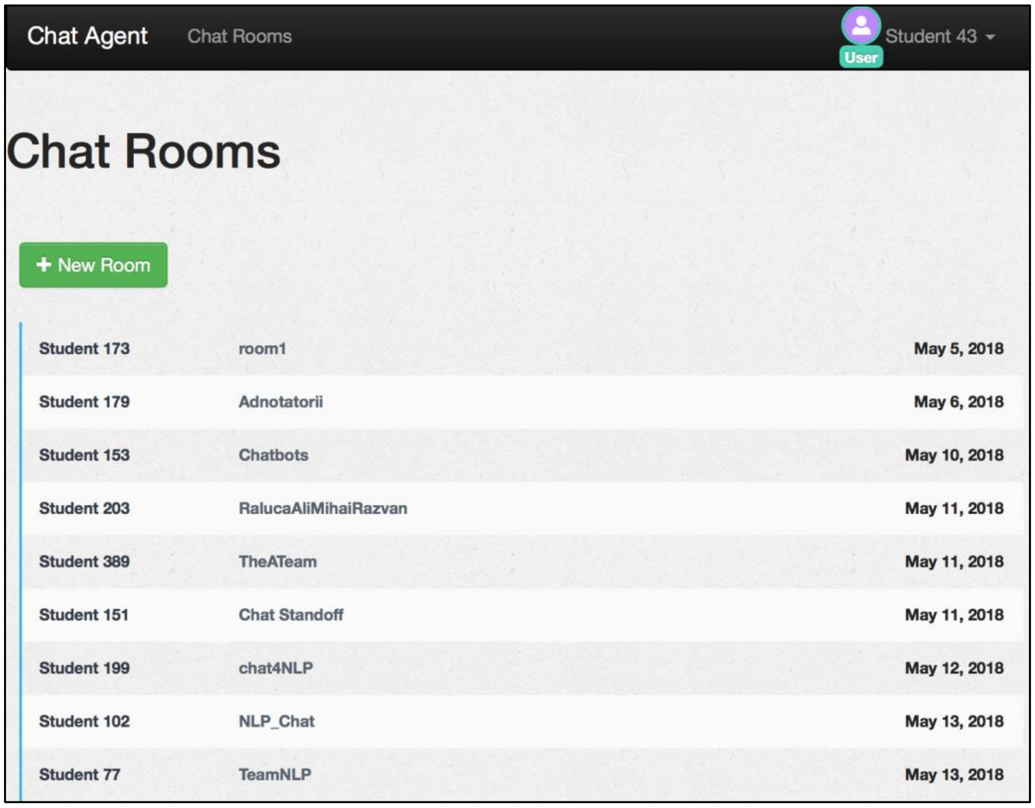
The chat application is part of the *ReaderBench* website [14] and is available at http://chat.readerbench.com. It works as a stand-alone application that can be used only by registered users. The homepage displays a guide that explains how users can register and login, how rooms can be created or how other conversations can be joined. Additional explanations regarding the interface of a room are also displayed, followed by short explanations describing the bots and the automated analyses perform on the conversations.

Our chat application is characterized by three main features: asynchronous messaging, automated bots for enhancing participants' activity and interactions, and the integrated NLP functionalities. Asynchronous messages allow users to access conversations, even after signing out and rejoining a room later on. Automated bots help participants to focus on the conversation by suggesting them to be more active or to express their opinions with regards other participants' ideas. Advanced NLP functionalities include visual graphs performed after a discussion is finished, with the help of the *ReaderBench* API. The graphs are based on the outputs provided by the *ReaderBench* processing endpoint.

The List of Chat Rooms

On sign-up, users have to read the Information Consent, which provides details that their contributions may be used for scientific research and that users have the right to be forgotten from the system; on this request, their usernames will be replaced by a random nickname.

Upon sign in, users are redirected to a page listing the discussion rooms. Here, they can either create a new room or join an existing one. Figure 1 shows the interface of the chat application displayed after sign in. Each conversation in the list displays its creator, the room name and the creation date. After clicking a conversation, the user is redirected to the room.



Chat Rooms

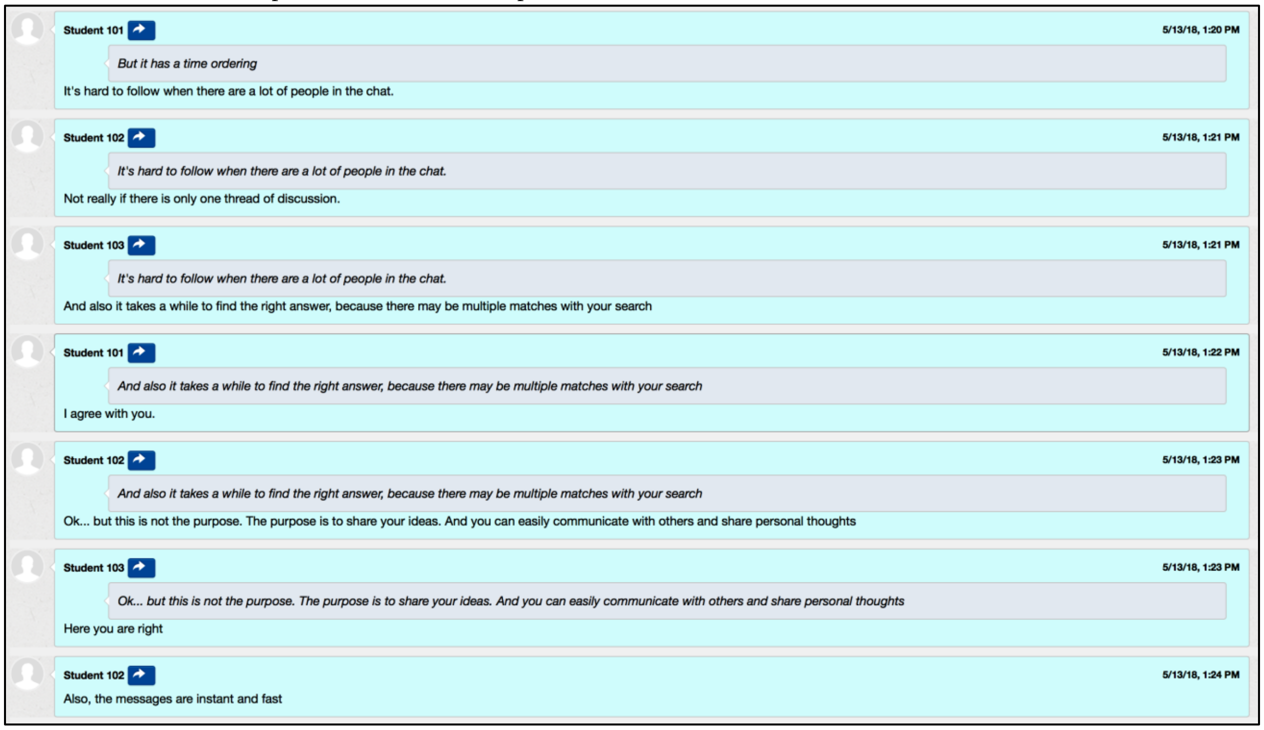
Student 173	room1	May 5, 2018
Student 179	Adnotatorii	May 6, 2018
Student 153	Chatbots	May 10, 2018
Student 203	RalucaAliMihaiRazvan	May 11, 2018
Student 389	TheATeam	May 11, 2018
Student 151	Chat Standoff	May 11, 2018
Student 199	chat4NLP	May 12, 2018
Student 102	NLP_Chat	May 13, 2018
Student 77	TeamNLP	May 13, 2018

Figure 1. Main interface after login showing all available chat rooms.

The Room Interface

Figure 2(a) shows part of a discussion established between a group of students within the chat application. It can be observed that some utterances were added as replies to previous utterances. This can be performed with the help of the "reply-to" button displayed as an arrow besides the participant's name. The application also saves the timestamp of each contribution and it exports discussions as an XML files, following the format which ensures compatibility to previous experiments.

(a)



Student 101 5/13/18, 1:20 PM

But it has a time ordering

It's hard to follow when there are a lot of people in the chat.

Student 102 5/13/18, 1:21 PM

It's hard to follow when there are a lot of people in the chat.

Not really if there is only one thread of discussion.

Student 103 5/13/18, 1:21 PM

It's hard to follow when there are a lot of people in the chat.

And also it takes a while to find the right answer, because there may be multiple matches with your search

Student 101 5/13/18, 1:22 PM

And also it takes a while to find the right answer, because there may be multiple matches with your search

I agree with you.

Student 102 5/13/18, 1:23 PM

And also it takes a while to find the right answer, because there may be multiple matches with your search

Ok... but this is not the purpose. The purpose is to share your ideas. And you can easily communicate with others and share personal thoughts

Student 103 5/13/18, 1:23 PM

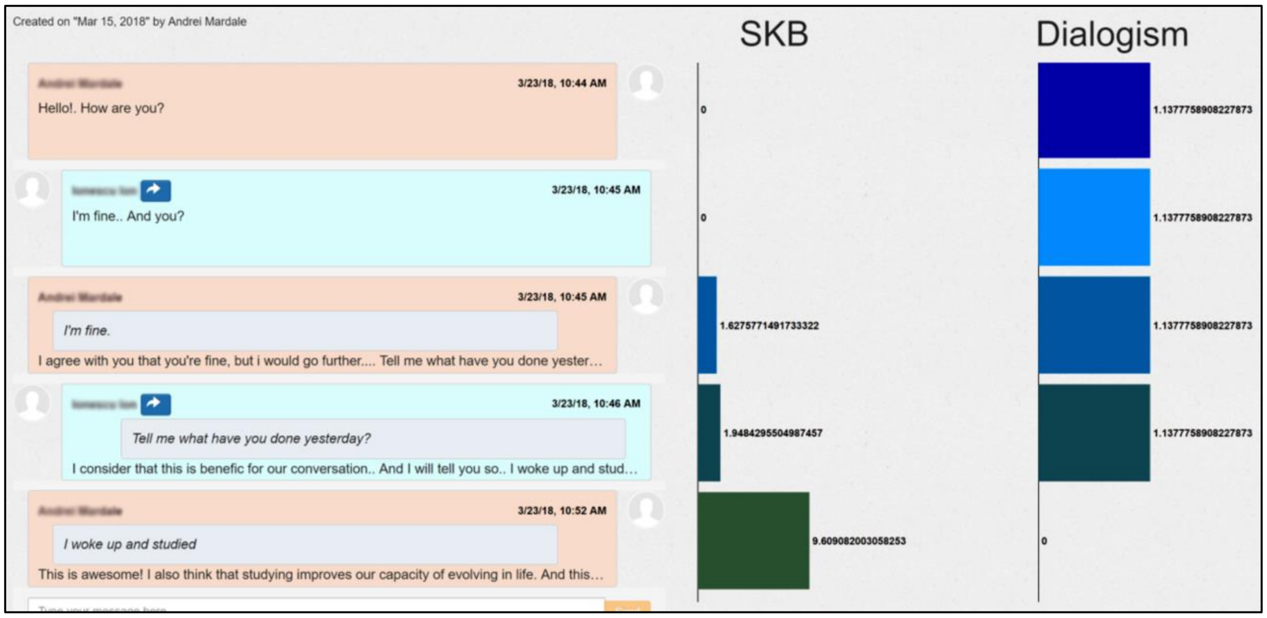
Ok... but this is not the purpose. The purpose is to share your ideas. And you can easily communicate with others and share personal thoughts

Here you are right

Student 102 5/13/18, 1:24 PM

Also, the messages are instant and fast

(b)



Created on "Mar 15, 2018" by Andrei Mardale

[redacted] 3/23/18, 10:44 AM

Hello!. How are you?

[redacted] 3/23/18, 10:45 AM

I'm fine.. And you?

[redacted] 3/23/18, 10:45 AM

I'm fine.

I agree with you that you're fine, but i would go further.... Tell me what have you done yester…

[redacted] 3/23/18, 10:46 AM

Tell me what have you done yesterday?

I consider that this is benefic for our conversation.. And I will tell you so.. I woke up and stud…

[redacted] 3/23/18, 10:52 AM

I woke up and studied

This is awesome! I also think that studying improves our capacity of evolving in life. And this…

Figure 2. (a) Preview of a discussion performed within the chat application.
(b) Social Knowledge Building and Dialogism evolution graphs of a sample conversation.

The Chat Bots

Within a discussion, the implemented chat bots intervene and try to regain the focus of users and to stimulate their participation. The current version of our application integrates two chat bots. The first bot is aimed at increasing users' interaction by asking participants, who did not formulate replies for other participants' contributions, to express their thoughts. The second bot tries to stimulate the conversation by asking users who did not show activity within a predefined time period if they are still present in the discussion or what is their opinion.

NLP Services

The chat application communicates with *ReaderBench API* and the response of the service is used for building visual graphs such as the social knowledge building and the dialogism evolution graphs [15] – see Figure 2(b). Figure 3(a) presents the concept map of a sample conversation, where the size of a node (i.e., keyword) shows its relevance throughout the conversation, while the length of an edge depicts the semantic distance between two concepts. Only the links with similarity scores above the provided threshold (which is set to 0.3 out of 1 in the application) are drawn. Figure 3(b) shows the participant evolution graph for the conversation, which helps monitor the evolution of each participant throughout the conversation. The horizontal axis follows the conversation timeline, while the vertical axis shows the overall score of user's contributions.

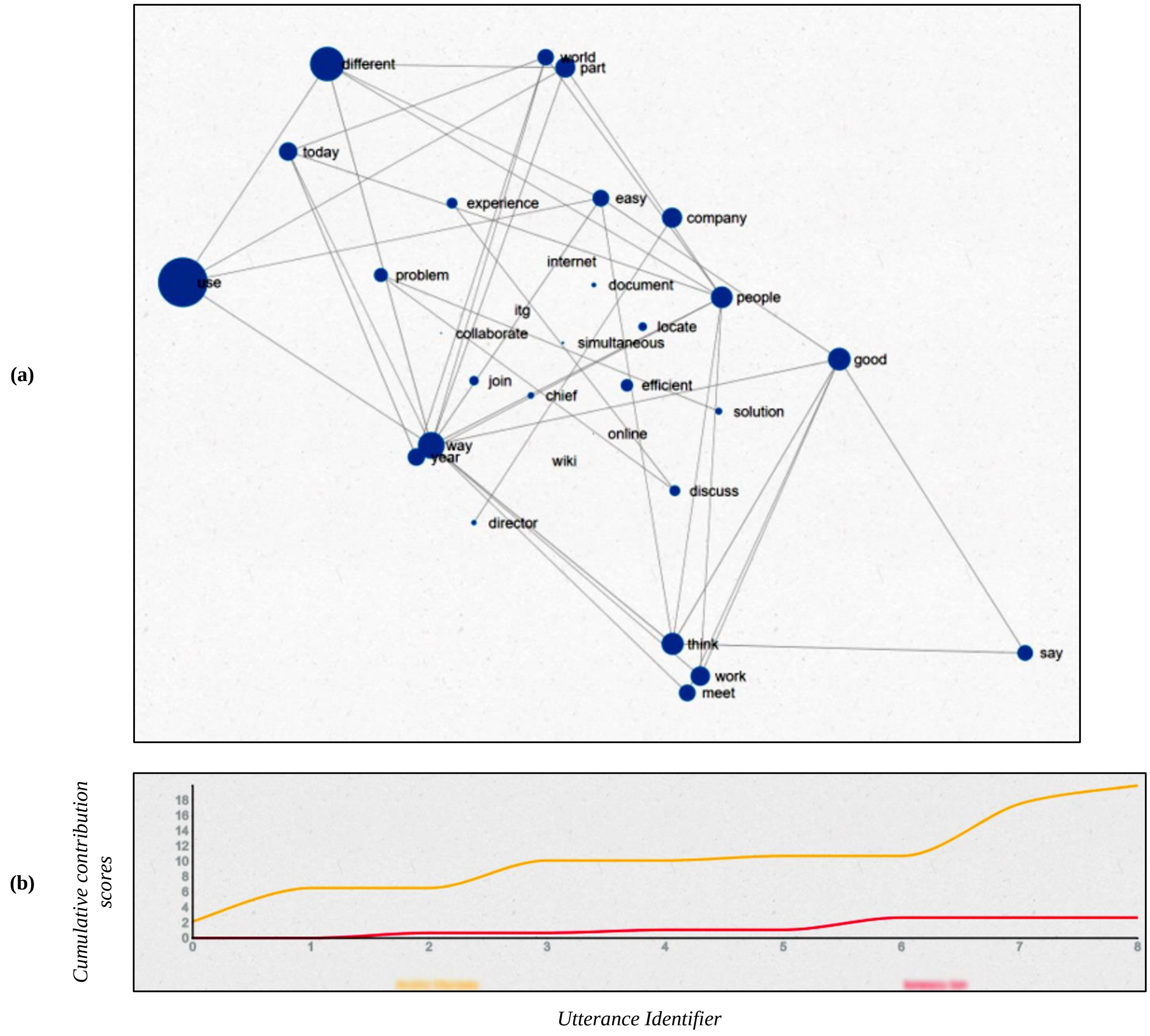


Figure 3. (a) Concept Map and (b) Participant Evolution graph for a sample conversation.

FEEDBACK

The students involved in our experiments were asked to provide free text feedback for the chat application by e-mail. Although the number of feedbacks was pretty small (just 6 out of about 40 participants), they helped us identify the strengths, the main issues and some recommendations for improving the application. Table 1 shows this information together with the corresponding number of students emphasizing a certain idea, in descendant order. Based on the gathered feedback, we evaluate the relevant issues or recommendations and propose solutions for them. We also assign priority scores (on a scale of high, medium and low), which are aimed at showing the most important issues to be solved by the developers.

Strengths

Two out of the six students who provided feedback mentioned the application's strengths. Both considered the application as a good environment for conversation (S1) which can be used with ease (S2). One user noticed that the application was able to maintain the conversation even after they accidentally closing the browser (S3), which is also considered positive. Students noticed the capability to use advanced NLP functionalities on the resulting conversation.

Issues

The students focused more on identifying issues and each of them emphasized at least two problems. The highest priority is assigned to the lack of the window's auto-scroll facility when new messages are received (I1), which was emphasized by four out of six students. Long duration in sending and receiving messages (I2), as well as misleading messages of users joining and leaving the room even when this did not happen (I3) was noticed by two of the users. One out of six students also noticed the inability to use the "Drag & Drop" facility for simulating the "reply-to" option (I4). Another student encountered problems with messages that were sent, but never showed up in the conversation (I5).

I1 should be immediately incorporated as it helps users to focus at new messages and easily follow the conversation. I2 was identified as being caused by the limited hardware resources currently being used for testing the application. The usage of this platform in a production environment with additional resources should be tested. I3 is a coding bug related to the calculation of the users' time of inactivity and it was immediately addressed. I4 was not observed during our internal tests on multiple operating systems and browsers, thus it might be a specific issue to the user's environment; follow-up analyses will be conducted. I5 also did not occur during our tests and, being reported by only one user out of 40, might exhibit a specific environment condition. However, since this is a critical functionality, we decided to consider this issue as more important. Based on these assumptions, we assigned priorities for each one of the identified issues.

#	Feedback	No. students	Priority
	Strengths (2 students)		
S1	NLP functionalities: The application is a good environment for chat conversations and NLP tasks related to them.	2 / 2	N/A
S2	Ease of use: The application is exciting, and it can be easily used by any user.	2 / 2	N/A
S3	Persistency: The conversation could be found even after the user accidentally closed the browser.	1 /2	N/A
	Issues (6 students)		
I1	The users had to scroll when new messages are received in order to read them.	4 / 6	High
I2	Lags in sending and receiving messages were observed.	2/ 6	Low
I3	Misleading messages of users joining and leaving the room were noticed.	2 /6	High
I4	The "Drag & Drop" facility for the "reply-to" option could not be used.	1 / 6	Low
I5	Some messages that were sent did not show up in the conversation.	1 / 6	Medium
	Recommendations (2 students)		
R1	An option to disable the bots should be included.	2 / 2	Medium
R2	Awareness messages (e.g., showing that a participant is typing a new message) should be included.	1 / 2	Low

Table 1. chat application issues, recommendations and strengths based on students' feedback.

Recommendations

Just two of the six students provided recommendations. Both of them requested to include a facility for disabling the bots (R1). One of them also proposed the inclusion of some awareness messages – e.g., a message showing when someone is typing a new contribution (R2).

With regards to the priorities of these recommendations, R1 might be helpful for users' interaction with the platform, but it is not beneficial to certain educational conditions that we want to setup. R2 might also be beneficial, but our experience with other applications showed that this kind of messages are displayed when only two participants are involved in a conversation; otherwise, the chances are extremely high that someone from a group of 5 persons is typing at a certain moment, thus making the message less helpful.

CONCLUSIONS

This paper introduced our new chat application – *Talk* – integrated with the *ReaderBench* framework. The application allows multiple users to connect to a chat room and discuss. It also provides multiple analyses making use of the *ReaderBench API* and the corresponding advanced NLP functionalities. The aim of this application is to support users discuss various topics and even solve tasks in a collaborative manner, encourage creativity, as well as active participation.

In order to test the environment, multiple students were asked to discuss on a specific topic specific to the course curriculum. Their conversations were exported and saved for future analyses. Students were also asked to provide feedback, which helped us identify the strengths, the issues, and some recommendations for future development. In general, students found the application as being useful for collaboration. The identified issues and the provided recommendations were mainly related to the interface and to users' interactions. Overall, students were able to solve the assignment without encountering severe problems. The provided feedback will be used to improve our application in subsequent releases.

ACKNOWLEDGMENTS

This work was partially funded by the 644187 EC H2020 Realising an Applied Gaming Eco-system (RAGE) project.

REFERENCES

1. Stahl, G., Group cognition. Computer support for building collaborative knowledge. MIT Press, Cambridge, MA, 2006.
2. Trausan-Matu, S., 2010. The polyphonic model of hybrid and collaborative learning. In Handbook of Research on Hybrid Learning Models: Advanced Tools, Technologies, and Applications, F. Wang, L., J. Fong. and R.C. Kwan Eds. Information Science Publishing, Hershey, NY, 466–486.
3. Manning, C.D. and Schütze, H., Foundations of statistical Natural Language Processing. MIT Press, Cambridge, MA, 1999.
4. Gutu, G., Dascalu, M., Rebedea, T., and Trausan-Matu, S., 2017. Time and Semantic Similarity – What is the Best Alternative to Capture Implicit Links in CSCL Conversations? In 12th Int. Conf. on Computer-Supported Collaborative Learning (CSCL 2017) ISLS, Philadelphia, PA, 223–230.
5. Trausan-Matu, S., Dascalu, M., Rebedea, T., and Gartner, A., (2010) Corpus de conversatii multi-participant si editor pentru adnotarea lui. Revista Romana de Interactiune Om-Calculator 3, (1), 53–64.
6. Holmer, T., Kienle, A., and Wessner, M., 2006. Explicit Referencing in Learning Chats: Needs and Acceptance. In Innovative Approaches for Learning and Knowledge Sharing, First European Conference on Technology Enhanced Learning, EC-TEL 2006, W. Nejdl and K. Tochtermann Eds. Springer, Crete, Greece, 170– 184.
7. Dascalu, M., Dessus, P., Bianco, M., Trausan-Matu, S., and Nardy, A., 2014. Mining texts, learner productions and strategies with ReaderBench. In Educational Data Mining: Applications and Trends, A. Peña-Ayala Ed. Springer, Cham, Switzerland, 345–377.
8. Dascalu, M., Analyzing discourse and text complexity for learning and collaborating, Studies in Computational Intelligence. Springer, Cham, Switzerland, 2014.
9. Trausan-Matu, S., Stahl, G., and Zemel, A., 2005. Polyphonic Inter-animation in Collaborative Problem Solving Chats. Drexel University.
10. Mühlpfordt, M. and Wessner, M., 2005. Explicit referencing in chat supports collaborative learning. Proceedings of the Proceedings of th 2005 conference on Computer support for collaborative learning: learning 2005: the next 10 years! (Taipei, Taiwan2005), International Society of the Learning Sciences, 1149353, 460-469.

11. Trausan-Matu, S. and Stahl, G., 2007. Polyphonic inter-animation of voices in chats. In CSCL'07 Workshop on Chat Analysis in Virtual Math Teams ISLS, New Brunwick, NJ, 12.
12. Dascalu, M., Gutu, G., Paraschiv, I.C., Ruseti, S., Dessus, P., McNamara, D.S., Crossley, S., and Trausan-Matu, S., 2017. Cohesion-Centered Analysis of CSCL Environments using ReaderBench. Proceedings of the 18th Int. Conf. on Artificial Intelligence in Education (AIED 2017) – Interactive Event (Wuhan, China2017).
13. Dascalu, M., McNamara, D.S., Trausan-Matu, S., and Allen, L.K., (2018) Cohesion Network Analysis of CSCL Participation. Behavior Research Methods 50, (2), 604–619.
14. Gutu, G., Dascalu, M., Trausan-Matu, S., and Dessus, P., 2016. ReaderBench goes Online: A Comprehension-Centered Framework for Educational Purposes. In Romanian Conference on Human-Computer Interaction (RoCHI 2016), A. Iftene and J. Vanderdonckt Eds. MATRIX ROM, Iasi, Romania, 95–102.
15. Dascalu, M., Trausan-Matu, S., McNamara, D.S., and Dessus, P., (2015) ReaderBench – Automated Evaluation of Collaboration based on Cohesion and Dialogism. International Journal of Computer-Supported Collaborative Learning 10, (4), 395–423.