Annals of the Academy of Romanian Scientists
Series on Science and Technology of Information
ISSN 2066 - 8562 Volume **4**, Number **2**/**2011**

AUTOMATIC LINKS IDENTIFICATION IN CHAT CONVERSATIONS

Andrei DULCEANU,[1] Ştefan TRAUSAN-MATU[2]

Abstract. *This paper considers one of the emerging branches of the science of learning, Computer-Supported Collaborative Learning (CSCL), together with an up-to-date issue of this field: identifying relationships among utterances in a chat performed by students. This issue is just a particular case of a more general one - determining discussion threads in a conversation on a CSCL-supported platform (forum, chat, etc.) - by using techniques of natural language processing and machine learning from artificial intelligence. Furthermore, several approaches of the problem, with the corresponding results, will be presented.*

Keywords: CSCL, polyphony, NLP, implicit links, WEKA, NLTK, TagHelper, chats

1. Computer supported collaborative learning

Computer supported collaborative learning is a "new branch in the science of learning which aims to study the way people learn together helped by computers" [19]. Although CSCL gained its popularity in recent years, the term was coined in the late 80s and the first projects and conferences were held in the mid 90s. Currently, CSCL uses modern tools for its purposes and the contributions of the research in this field are reflected in various sciences like software industry, education, psychology or sociology. Thus, the classical process of learning is improved towards a modern, transparent and interactive method which brings the advantages of collaborative learning [12]:

- positive mutual dependency: participants are aware that they depend on each other and that they need each other to accomplish the task
- individual responsibility: responsibility is divided among group as a whole and its individual members
- social abilities: in order to work in such a group, one needs social abilities as well as communication abilities.

Unfortunately, studies have shown that this flavor of learning facilitated by computers can have a negative impact on the interaction between group members. "It was observed that in these groups trust, cohesion, efficiency, as well as the change of ideas are suffering" [14] which affects the overall efficiency.

[1]M.Sc., Faculty of Automatic Control and Computers, University Politehnica of Bucharest, Bucharest, Romania (andrei.dulceanu@gmail.com).
[2]Prof., Ph.D., Faculty of Automatic Control and Computers, University "Politehnica" of Bucharest, Romania; corresponding member of the Academy of Romanian Scientists (stefan.trausan@cs.pub.ro).

Moreover, tutors are focusing too much on the technologies which facilitate the interaction, neglecting the process itself, especially because of the lack of documentation on the subject.

Artificial Intelligence (AI) tries to surpass these shortcomings by combining machine learning techniques with Natural Language Processing (NLP) techniques in order to produce richer views and outputs about chats between students, shared information or feedback based on a group investigation model. Therefore, sociability is supported through monitoring interaction models [19]. One of the main problems of using NLP for processing chats is the detection of links between utterances and their grouping in threads. This paper presents and analyzes three experiments for detecting such links.

The paper continues with a section analyzing some state of the art approaches in the analysis of links in chats. Section 3 introduces the considered problem while section 4 describes the experiments. The last section introduces some conclusions.

2. Related work in thread detection

In order to have some terms of comparison for our three experiments, we will describe three previous main categories of research in automating the analysis of CSCL artifacts: thread detection in chat conversations, forums and email groups, email summarization and collaborative process analysis.

2.1. Thread detection in chat conversations, forums and email groups

Detection of threads in discussion forums is usually considering the explicit "reply-to" linking of messages. One approach for determining threads in forum conversations, without using meta-data like "posted by" or "reply to" uses the collection of messages represented as a graph, the edges being the similarities among messages [27]. The authors state that consensus discovery, knowledge building and adjacency pairs discovery for sequential relevance identification represent advantages of determining the threads in a conversation. The corpus used in the experiments was generated by an educational collaborative tool and does not contain any reference about the relation between utterances. The data was represented as term arrays weighted by their (tf) and their inverse document frequency (1/df, df being the number of messages containing the term), considering also the length of the messages (msglength):

$$TF.IDF = \frac{tf}{tf + 0.5 + 1.5 \cdot \frac{msglength}{avg\_msglength}} \cdot \log\left(\frac{N}{df}\right), \text{ N = no. of messages} \qquad (1)$$

Most typical terms (from a similarity point of view) are highlighted by weighting the terms (1). Each message in an ordered message sequence is considered a

vertex in a graph, each element of adjacency matrix $W = [w_{ij}]_{n \times n}$ defining the weight of the edge between m_i and m_j (2). W represents the matrix of semantic similarity between messages, m being the vector with the number of apparitions of each term in a message.

$$w_{ij} = \begin{cases} \dfrac{\vec{m}_i \bullet \vec{m}_j}{\|\vec{m}_i\| * \|\vec{m}_j\|}, & i > j \\ 0, & otherwise \end{cases} \quad (2)$$

Finally, the structure of the threads is rebuilt by using a parent-child decision threshold between messages. If their similarity is greater than the threshold, then a new edge (3) will be inserted in the discussion threads graph $G = [g_{ij}]_{n \times n}$.

$$g_{ij} = \begin{cases} 1, w_{ij} > threshold \\ 0, \quad otherwise \end{cases} \quad (3)$$

Apart from the basic algorithm (GRB), the authors propose three other flavors which replace the *W* matrix with another one built on the following observations: only messages in a certain time window can be considered parents for other messages (GRF), dynamic definition of the time window in the former approach (GRD), message similarity is weighted using their distance in time (GRT).

Experiments have shown that GRF and GRT algorithms beat the basic algorithm with almost 30% and GRB beats GRD with 28% which comes to confirm the hypothesis saying that child-messages are close to parent-messages.

Another approach for thread detection, based on graphs of utterances in chats considers as arcs both the explicit links (indicated by the users of the chat environment ConcertChat [9], used in the experiments) and implicit links (detected by natural language processing techniques (like adjacency pairs, repetitions, argumentation links) connecting the utterances, which are the nodes of the graph [24]. Threads are chains of arcs which behave like a "voice" in a generalized way, for example, chains of repetitions, lexical chains or argumentation chains.

2.2. Email summarization

Email summarization may be considered a big challenge when it comes to the need of the users to organize their emails with anti-spam filters and to classify and view them, given the fact that the number of emails one receives has expanded dramatically [5]. Moreover, mobile devices make it even more important due to the need of summarized presentation of the content.

An approach for summarization [5] is to build a graph of quoted excerpts for discovering clue words which summarize a message. A word in the excerpts

graph is a clue word only if it appears both in parent nodes as well as in their children nodes, having the same meaning. The excerpts graph G=(V,E) is a oriented graph in which each vertex is a text fragment from a message and an edge between vertices u and v (u -> v), denotes an answer from u to v. Excerpts are represented by continuous lines beginning with a separator (e.g. ">") and can be shuffled on various levels with new excerpts or missing excerpts.

Edges between vertices are added by noticing that "each new excerpt might be a potential answer for quoted excerpts in its neighborhood. The authors propose CWS (ClueWordSummarizer) algorithm which represents the importance of each word with *ClueScore(CW,F),* a function which takes as parameters the clue word CW, the excerpt F,

$$ClueScore(CW,F) = \sum_{parent(F)} freq(CW, parent(F)) + \sum_{child(F)} freq(CW, child(F)) \quad (4)$$

and needs the frequency of the clue word in the excerpt (freq). The score for a whole sentence in the excerpt is computed as follows:

$$ClueScore(s) = \sum_{CW_i \in s} ClueScore(CW_i, F) \quad (5)$$

In order to generate a summarization for *k* sentences, the CWS algorithm tokenizes each sentence from each node, resulting a word multiset. Stop-words are stripped off. Words are then stemmed and after that the above formulas are applied on words and sentences. The output is represented by selecting the first k sentences based on the highest score.

2.3. Collaborative process analysis

CSCL argumentative knowledge building analysis may take into account dimensions like: epistemic activity, micro-argumentation, macro-argumentation, co-building social modes, reaction, feedback to learning environment hints, and discourse quotation [28].

Polyphony, a concept taken from music theory may be considered in order to model the interaction in a group of chat users similarly to a mix of voices (tunes) which develop various topics unified by a central main topic (harmony) [24].

Inter-animation patterns in a chat conversation may be classified as *adjacency pairs* – pairs of logical grouped successive utterances (e.g. question-answer pairs), *repetition* – associated to rhythm in musical theory, *cumulative/collaborative utterances* – several users contribute to a complex topic as one and *convergence* – utterances which unify two different conversational threads in order to assure harmony [24].

An application was developed for analyzing chat conversation logs [23]. One feature of the application aims to identify the topics in a chat conversation by tokenizing the input and unifying similar concepts based on WordNet lexical ontology (http://wordnet.princeton.edu). The proposed strategy is improved by an empirical one which takes into consideration several cue phrases for introducing new topics in a conversation (e.g. "let's talk about <>", "what do you think about <>", etc.).

A similar approach was taken for identifying implicit links between chat utterances. Linguistic patterns are used to find referenced words and then a search in an established time window is performed for marking utterances in which referred words appear as implicit links.

3. Problem description

There is a difference between face-to-face and online conversations [1]. Chat users tend to use shorter sentences, to focus more on the environment than on the topic and to switch topics too often. This leads to conversations hard to follow. It is therefore a necessity to have tools which can help the persons who analyses chats to focus on a single topic by aggregating, organizing and informational evaluating their utterances. This need generated many directions for developing applications in the CSCL community, like: solving problems using chat conversations, grading and evaluating students in a chat conversation, comparing students' knowledge in a group, chat conversations summarization, and detecting implicit links in a chat conversation.

In order to identify links between utterances in a chat conversation we need to take into consideration the above conclusions referring to topic diversity in a chat conversation. A thread in a chat conversation is represented by the development of a topic from various utterances. An implicit link between two utterances is represented by an utterance which refers to another utterance in the same thread. Trausan-Matu & Rebedea [25] suggest that a correct analytical approach of the links and threads in a conversation is given by coherence and cohesion found in that conversation. Moreover, Fulks & Pimentel [7] consider four new approaches for identifying implicit links: message posting analysis, topic analysis, context analysis and conversational sequences analysis combined with users' utterance variation.

Going back to coherence relations we can have the following types of implicit links [30]: cause-effect, condition, immaterialized assumption, similarity, contrast, temporal sequence, quotation, example, elaboration and generalization. In a paper about email conversations summarization [5], repetitions of similar words are analyzed from a conversational cohesion point of view. In NLP this is known as a lexical chain. A lexical chain is a semantically akin word sequence which can be found in contiguous sentences sequences in a document. Lexical chains are mostly

used for conversation summarization, but can be used also for identifying implicit links because they consider only the meaning of the words in an utterance and not conjunctions or any other stop words.

The approach considered in this research was to automatically identify several types of links/speech acts/patterns in a corpus of chat conversation as follows: speech acts in Experiment 1 and Experiment 2, implicit and explicit links between chat utterances in Experiment 2, and inter-animation patterns in Experiment 3

4. Experiments description

The corpus of conversations used for training and testing consisted of dialogs which implied the use of the concertChat chat client [9] for debating over pre-assigned topics. All students that took part in these chat sessions were enrolled in the Faculty of Automatic Control and Computers, University ”Politehnica” of Bucharest. The chat sessions were course assignments and were graded accordingly.

ConcertChat is a complex Java chat client which includes features like whiteboard – for interactively drawing figures, images, etc. - and explicit referencing of utterances or parts of them.

4.1. Experiment 1: Speech acts identification using TagHelper

The TagHelper System and data modelling

In the first experiment we used TagHelper, a system for “facilitating reliable content analysis of corpus data” as it is described by its authors [15]. TagHelper increases classification performance by taking into account seven features: punctuation, unigrams and bigrams (words or adjacent pair of words), POS (Part-of Speech) bigrams - pairs of grammatical categories, utterance length, stop-words, word-sense disambiguation and rare words.

TagHelper takes as input Microsoft Excel spreadsheets because of their wide adoption among scientific community for data analysis tasks. The file should have a standard format, respecting three major types of columns.

The first and the most important column is “Text” and it cannot be skipped from the file; it contains the utterance per se to be analyzed. Before “Text” we have the columns to be considered dimensions in the coding scheme; every question mark will be treated as not coded on the corresponding dimension and will be coded by TagHelper using the trained model obtained on the annotated examples. Additional columns placed after “Text” column are treated as “extra features” to be used in the predictive model. The experiment in this paper uses four extra features: “Explicit reference”, “Id”, “Name” and “Date”.

In order to be processed with TagHelper the xml conversations were transformed in Excel files which define the following columns:

- *ID:* corresponding to the *genid* attribute in the *xml* file, it represents the unique identifier of an utterance in the conversation
- *NAME:* corresponding to the *nickname* attribute, it represents the nickname used by the user in the chat session
- *DATE:* mapping from *time* attribute, it stores the complete date and hour for the current utterance
- *CONTENTS:* mapping from *Utterance* tag, it contains the utterance per se
- *EXPLICIT REFERENCE:* mapping from *refid* attribute, it represents the ID of the utterance referenced by the current utterance
- *REFERENCE:* referenced word (e.g.: „him")
- *ARGUMENTATION:* reference type (e.g.: condition, see below)
- *SPEECH ACT:* speech act type (see below).

Speech acts theory

Austin [2] introduced speech act theory remarking that some utterances in a dialog play the role of actions executed by the speaker [10]. Austin's theory is applicable to performative verbs as well as to other classes of verbs, and it considers that each utterance may contain up to the following three categories of acts:

- Locutionary act: the utterance of a sentence with a particular meaning
- Illocutionary act: the act of asking, answering, promising, etc., implicitly included in the locutionary act
- Perlocutionary act: the (often intentional) production of certain effects upon the feelings, thoughts or actions of the addressee in uttering a sentence [10].

Searle [17] modifies Austin's taxonomy dividing speech acts into five classes:

- Assertives: the speaker is directly committed to a certain action
- Directives: in which the speaker asks the addressee to do something
- Commissives: in which the speaker commits to some future course of action
- Expressives: expressing the speaker's psychological state about a certain situation
- Declarations: a new state is introduced via the utterance.

Speech act theory was extended by Bunt [4] for adding information about adjacency pairs, grouping contributions or notions, consequently introducing the concept of dialogue act. For a standardized labeling of dialogue acts DAMSL (Dialogue Act Markup in Seven Layers) architecture was created for "coding on different levels the information which resides in utterances in a dialog" [21].

The 18 labels being used in the experiment for the "SPEECH ACT" column of the TagHelper input spreadsheet are not the subject of this architecture. They were chosen from [10], with their corresponding examples from a total amount of 43 labels. The last two labels were added by me for a better coverage of the annotation scheme and the examples provided were chosen randomly from the annotated corpus.

Table 1

Labels associated with the SPEECH ACTS column

Label	Example
Thank	Thanks
Greet	Hello Dan
Introduce	It's me again
Bye	All right bye
Request-comment	How does that look?
Suggest	from thirteenth through seventeenth June
Reject	No, Friday I'm booked all day
Accept	Saturday sounds fine
Request-suggest	What is a good day of the week for you?)
Init	I wanted to make an appointment with you
Give-reason	Because I have meetings all afternoon
Feedback	Ok
Deliberate	Let me check my calendar here
Confirm	Ok, that would be wonderful
Clarify	Ok, do you mean Tuesday the 23rd?)
Digress	[we could meet for lunch] and eat lots of ice cream]
Motivate	We should go to visit our subsidiary in Munich
Garbage	Oops, I…)
Completion	Additionally I think we can find a new solution...
Repeat-rephrase	Wikis are indeed a good tool and are also suitable for our project

A new way of identifying links between utterances was introduced [8, 11], which will be used for the "ARGUMENTATION" column representing a super-type of the labels used in "SPEECH ACT" column.

The algorithm used in TagHelper [15] for learning was Naïve Bayes [13].

Table 2

Labels associated with the ARGUMENTATION column

Category	Discourse markers	Example
Reason	because, since	*Since* we rely on the supply from the mainland…
Condition	If	*If* we don't have enough land
Consequence	then, thus, so, therefore	*So* why don't we find some places where has more land…
Contrast	but, though, although, however, even, otherwise	*Although* some of the pollutants blocked out some sunlight…
Elaboration	moreover, such as	*Moreover,* we can use reusable energy such as wind power
Claim	I think, I agree, We should	*I think* that HK Government should set up laws and make a random inspection
Question	What, Why, How	*Why* can spaceflight help humans develop a better world?
Answer	Yes, No	Yes, this is true.
Rebuttal	"I don't think", "I don't agree", "we shouldn't"	*I don't think* budgets are serious problem.
Cohesion	Also, besides	CO2 can be produced with chemical method; *also,* we may have new technology…

Conclusions and results

Regarding conclusions, few words shall be mentioned about the corpus used for training and the one used for testing. The former consisted of six files with dialogs in English summing up 2408 utterances, while the latter had three files summing up 1215 utterances. There were cases in which certain utterances were not linked to previous ones; hence all three columns were labeled "N/A" – Not Applicable.

Firstly, it should be noted that the precision of this experiment (~43,25%) must be considered of a greater importance than its recall (~23,70%), because the conversations from concertChat could not be treated as a whole. The reason was that, for those conversations in which the referred utterances were quite a few, the un-referred utterances were labeled too, therefore the model learned after training was disturbed by the un-referred, but annotated utterances which were not in a vast amount in the test corpus.

Secondly, the results for precision and recall metrics presented in the Tables 3 and 4 were influenced by a limitation of TagHelper system which does not parse multiple values for a column, but considers them a whole. As an illustration, the annotation consisting of three different labels "confirm, deliberate, suggest" is used as a "block" in the test corpus providing rather poor results. Tables 3 an 4 present precision and recall metrics for the "SPEECH ACT" and "ARGUMENTATION" columns (Labels: *Thank*, *Greet*, *Introduce*, *Bye*, *Init*, *Feedback*, *Clarify*, *Digress* and *Garbage* were used for rather few utterances or were not used at all in the correct coding of test corpus; therefore they are not relevant for this experiment).

Table 3

Precision and recall for SPEECH ACT column

Label	Precision	Recall
Request-comment	85%	26%
Suggest	56%	73%
Reject	35%	31%
Accept	46%	64%
Request-suggest	100%	18%
Give-reason	42%	23%
Deliberate	0%	0%
Confirm	44%	48%
Motivate	60%	43%
Completion	21%	6%
Repeat-rephrase	N/A	0%

Table 4

Precision and recall for the ARGUMENTATION column

Category	Precision	Recall
Reason	N/A	0%
Condition	N/A	0%
Consequence	100%	12%
Contrast	N/A	0%
Elaboration	57%	31%
Claim	75%	4%
Question	93%	30%
Answer	8%	23%
Rebuttal	43%	42%
Cohesion	38%	57%

Thirdly, it was observed that this method cannot determine the referenced word. The values predicted for the "REFERENCE" column in the test corpus are filled up with words learned in the training phase which have nothing to do with conversations in the test file.

In order to have a clear picture about the results obtained, a comparison to the results obtained in a previous system [21] was performed. The approach used in the mentioned paper is based on heuristics, grouping speech acts in two categories ("forward looking function", "backward looking function"), recognizing verbs in utterances and searching for "cue phrases" for certain speech acts. The test corpus for that experiment consisted of three chats summing up 1200 utterances and the obtained results are presented in Table 5.

Table 5

Precision and recall for the previous system [21]

Category	Precision	Recall
Statement	92%	79%
Info_request	92%	92%
Action_directive	67%	69%

4.2. Experiment 2: Speech acts identification using NLTK

This experiment used the same corpus as the previous one, with manual annotated conversations, but only the SPEECH ACTS category was used here. The ARGUMENTATION column was stripped off in order to test the hypothesis in which ambiguous annotation for this column influenced the final results. Preprocessing included contraction expansions (e.g. *I'll* becomes *I will*, *won't* becomes *will not*), utterance conversion to lowercase, tokenization and lemmatization (WordNet was used).

A *bag of words* approach (in which each word in a sentence is considered a feature, without considering any order of appearance) was used for each utterance, generating features consisting of pairs *(word, True)* for each *word* appearing in the utterance. Difference classifiers have been tested and the best results were obtained with a maximum entropy classifier (Maxent classifier).

Results were compared with the ones obtained in the previous experiment and an improvement was noticed thanks to implicit links discovery among some of the utterances. Identification of the links was possible by using speech acts heuristics in which question-answer or completion utterances were grouped together.

The challenge was to surpass discourse changes in chat conversations, where people tend to produce a constant flow of messages, thus uttering complex sentences, very hard to categorize in terms of using only one speech act. The annotation scheme was changed in order to produce a more consistent model, respecting the following guidelines:

- short and medium utterances matching two or more categories were associated the speech act corresponding to the strongest discourse marker found in the utterance.
- long or very long utterances which present/reject/analyze/complete ideas were annotated with *elaboration*
- irony, jokes or utterances which aren't discussing about the main topic in the conversation were treated as *garbage*
- every answer to a question was considered to match *answer* category, although the utterance could be annotated as *elaboration, completion, etc.*, in order to underline links of type question-answer
- administrative concertChat utterances (e.g. *user has joined the room* or *user has left the room*) were stripped off because they didn't contribute at all to the overall conversation
- utterances in Romanian were deleted.

The Open-source Natural Language Toolkit (NLTK, http://*www.**nltk**.org)* was used for this experiment. ConcertChat transcripts are converted to xml files containing the utterance per se, the complete date and hour when it was typed, its author, its unique id and the id of the referenced utterance, were applicable. The xml files were converted to Excel files, each of the meta-data being mapped to a column. A new column was introduced to handle the speech act associated to the utterance.

After manual annotation of the corpus consisting of 6 conversations, each utterance was processed as follows:

1. The utterance was converted to lowercase.
2. Contractions such as *won't, can't, I'm, 'd, 'll* were expanded to *will not, cannot, I am, would, will*. This step was needed for explicitly marking modals and negations.
3. Tokenizing: the utterance was tokenized into valid words, stripping off most of punctuation, but keeping *?, !,), (* as independent tokens for identifying utterances corresponding to *question, rebuttal* or *garbage* categories.
4. Lemmatizing: each word was replaced by its lemma using WordNet, reducing derivate words to their roots and thus increasing correct classification probability.
5. Each token in the list was mapped to a feature consisting of the pair (token, True) respecting the *bag of words* model.

Although stemming and stop-words removal were tried, the results got worse and therefore these well-knows NLP processing techniques were aborted. Porter stemmer from NLTK was used for the former, independent or combined with lemmatizing, while the latter deleted important discourse marker for *reason* and *contrast* categories (e.g. *because, but).*

For example, from the initial utterance „*Indeed, but not every time you need detailed information"* the following features were generated:

{'detailed': True}, {'information': True}, {'indeed': True}, {'but': True}, {'need': True}, {'every': True}, {'time': True}, {'not': True}, {'you': True}

Different classifiers were trained and tested against the corpus and the maximum entropy classifier won by far. This classifier codes labeled features to vectors and then uses them to compute weights associated to each feature. These weights are combined in order to determine the label for which the probability is the highest in each given set [3].

Of all available NLTK implementations our choice was to use the gradient approach (recommended by the authors of the toolkit). Thirty iterations with variable weights were used. The number of iterations was determined empirically.

Implicit links identification heuristics

In order to identify implicit links between chat utterances, the *window_size* parameter was defined to be the maximal previous utterances number to consider as referenced candidates for the current utterance. The parameter was computed as the maximum of the distances between two utterances marked with an explicit link in concertChat, in all the four conversations considered for training the classifier. The value obtained was 23.

The first type of implicit link considered is question-answer pairs. After analyzing the conversations in the corpus, it was observed that, although a naïve approach, answers tend to respond to questions in the immediate vicinity.

Secondly, it was noticed that chat users are often using interlocutors' names or part of name. This is true for consensus situations: *I agree with Ionut*. The strategy used to identify this type of links is to start with utterances annotated with *confirm* and to search for candidate utterances not annotated with *introduce*, *garbage* or *question*, keeping the condition that current utterance must contain part of the candidate utterance author's name.

Thirdly, the algorithm tries to cover the cases in which users complete previous utterances – their own utterances or other participants' utterances. A similarity measure is used between current utterance and candidate utterance which takes into account distance in time between utterances, annotated speech acts and authors' names. Similarity is computed as the mean of the above criteria, in which consecutive utterances in a short time window submitted by the same author and with the same speech act are advantaged.

Results and conclusions

The first part of the experiment used a training corpus comprising four conversations and a total number of 1181 utterances and a test corpus comprising two conversations with a total number of 704 utterances. The accuracy of the classifier was 37.64% and the individual measures for each of the speech acts are presented in Table 6:

Table 6

Precision and recall for each speech act

Speech act	Precision	Recall
Reason	25%	80%
Condition	46%	46%
Consequence	61%	53%
Contrast	9%	28%
Elaboration	23%	36%
Claim	50%	16%
Question	94%	92%
Answer	15%	28%
Rebuttal	18%	34%
Cohesion	23%	28%
Confirm	57%	59%
Introduce	84%	68%
Garbage	11%	23%

The first conclusion drawn after this experiment was that it out-performs the first experiment in terms of recall. Moreover, three categories not discovered in the former experiment – *reason*, *condition* and *contrast* – were successfully classified with the current method, producing good numerical results. The annotation method was greatly improved by adding three new categories and by using a more consistent scheme.

Turning to the heuristic algorithm which tries to discover implicit links between the utterances by taking into account the speech act associated with each sentence, it was run on a manual annotated conversation in order to take advantage of the higher accuracy of the manual annotation over the automatic one. It discovered 36 links, of which 18 were explicit (marked in concertChat) and 18 were implicit (not found in the training file). There were 145 explicit links in the conversation used for training. The final results for precision and recall of link discovery algorithm are presented in Table 7.

Table 7

Precision and recall for identified links

Link type	Precision	Recall
Explicit	61%	12%
Implicit	72%	N/A*

*total number of implicit links wasn't counted due to its order of magnitude

4.3. Experiment 3: Detecting inter-animation patterns in chat conversations using WEKA workbench

Training conversations were manually annotated, a single inter-animation pattern being associated with an utterance. Each chat was converted to Excel format and only pairs of referenced utterances were taken into consideration. Preprocessing included converting the utterance to lowercase, tokenization, and stop-words removal. The features considered were common words in the two utterances, the length of the original utterance, the length of the referenced utterance and four measures, one for each category in the set {question, contradiction, argumentation, negation}. Each of the measures is a Boolean measure which gives the affinity to a category in the set. The four measures were computed by analyzing the words in the utterance and by identifying some keywords for each of the categories.

The open-source workbench WEKA (http://www.cs.waikato.ac.nz/ml/weka/), used for the practical implementation of the system, includes several machine learning algorithms. They can be directly run against corpus data available in some WEKA-friendly formats (ARFF, CSV, etc.). Filters are available for preprocessing and tools for data conversion can transform input corpora into many formats [29].

A Java module was written for building the test corpus in ARFF format (one of the formats supported by WEKA). The features found/computed were comma separated written, one row for each pair of utterances. SMO classifier was used, which implements the minimal sequential optimization algorithm for training a SVM classifier using Gaussian or polynomial kernel functions.

Hybrid learning. Polyphony

Polyphony, a concept taken from music theory, is defined as a "general interaction and creativity model in a human "voices" group which respects counterpoint rules" [22, 26]. This concept is characterized by a central topic developed by several voices – affected or not by conflicts resolved without breaking the overall harmony – and by short term variations which keeps the coherence of the discourse.

Hybrid learning combines classical learning methods in the classroom with modern methods which target computers. Because chat is the most frequently used tool inside the CSCL community – among forums, email and chat – polyphony needs to be redefined in the new context of hybrid learning, when the starting point is a chat conversation between several participants. Trausan-Matu [26] considers hybrid learning "a polyphony of contributions pertaining to several participants, teachers or students, expressed through different types of utterances". A voice is assimilated to "a distinct position in a group, a person or group of persons which uttered specific utterances, affecting subsequent utterances".

Inter-animation patterns

Inter-animation patterns which appear in a chat conversation can be divided in two classes: unity and difference. The first category patterns contribute to the coherence and continuity of the discourse.

The first sub-class of this category is represented by adjacency pairs [16] which include question-answer utterances, greetings or, in a broader classification given by [18], proposal-accept and proposal-rejection pairs:

adjacency Ionut What do you represent tonight?
adjacency Cristi I'm going to present you the benefits of a blog solution

Excerpt 1. Question-answer adjacency pair

adjacency Ionut Hi Cristi
adjacency Cristi Hello, sorry i'm late, had a few technical problems

Excerpt 2. Greeting-greeting adjacency pair

adjacency Ionut so...I think that it is pretty hard to follow a conversation
adjacency Cristi Yes, that is correct, and also a lot of information is wasted

Excerpt 3. Proposal-accept adjacency pair

adjacency	**Ionut**	**I think that in a chat conference if there are many people with different ideas you can get to a point when you cannot follow the chat**
adjacency	**Cristi**	**you can have an archive :) but if you need to talk something very important very fast? if u need support for something? There are various applications for a chat**

Excerpt 4. Proposal-rejection adjacency pair

The second sub-class considers *repetitions* – which usually appear more often than adjacency pairs [26]. Repetitions are responsible with the rhythm of the conversation and for consolidating conversation topics. Besides the classical repetition of words of parts of previous utterances, there is another kind of repetition which was analyzed, in which words in a previous utterance were replaced with their synonyms in future utterances:

repetition	**Ionut**	**Actually a wiki has admins too, that generally verifies the *data* that is registered by users and organizes it so it can be *accessed quickly***
repetition	**Ionut**	**Wiki mean *fast* in Hawaiian language, from here we understand that this technology is one that is permitting *access* to information very easy and *fast***

Excerpt 5. Repetition (words in italics denote repetition)

The next sub-class is represented by *argumentation links*, in which users defend their point of view, clarify their ideas or answer to questions with solid motivations (easily mistaken with *adjacency pairs*):

adjacency	**Mihnea**	**and why is that, precisely?**
argumentation	**Mihai**	**because it's easy to post a message - so, easy to generate**
	Angela	**it took you 10 minutes to read what I said**
argumentation	**Cristi**	**the downside to a wiki is that you need to have a pretty good grip on the site otherwise you risk getting a lot of spam and very few useful information**

Excerpt 6. Argumentation links

Finally, the last sub-class is given by patterns which help to improve the coherence of the discourse: *convergence*. This pattern has a great importance given the fact that harmony of the discourse is directly affected by it; different voices present in the conversation are "melting" into a single voice. For example, Angela, Mihnea, Mihai and Ionuţ which defend a different technology between blog, wiki, forum or chat, draw the same conclusion about a complex product which can integrate each of their technologies. Consensus is established:

convergence	**Mihai**	**if I'm not mistaking some courses in upb integrated the facilities of wiki and forum**
convergence	**Angela**	**yes, and students are chatting, and have blogs**
convergence	**Mihnea**	**yes... it could be very useful to trade infos about homework issues, exams tips and tricks, have some wiki pages about all the infos you come across on different subjects, or about application process or find something useful about your teacher**
convergence	**Mihai**	**maybe adding live chat for quick help in desperate times (such as the last few hours of a deadline, we all know most of us are online at that time)**
convergence	**Ionut**	**Chatting all the time actually, that would be a good idea**

Excerpt 7. Convergence

The second category of patterns (difference making patterns) contains a single class, *contradictions*, in which chat users manifest rebuttal to ideas in previous utterances. In order to not overlap with utterances considered *adjacency pairs*, only utterances with a strong negative meaning were included in this class, and not partial rebuttals:

adjacency	**Anegla**	**yes, but is also very subjective**
difference	**Ionut**	**I consider a blog a "private space" *and not* a source of trustful information**
	Lili	**I say we move this on a blog... that way we take advantage of everybody's opinion**
difference	**Mihnea**	**you can't do that, once wikipedia is actually a self proclaimed encyclopedia**

Excerpt 8. Differences

After the manual annotation of the files in the corpus, each pair consisting of an original utterance and a referred utterance, the following steps were followed for each pair:

1. Utterances were converted to lowercase.
2. Tokenizing: utterances were divided into tokens, stripping of punctuation
3. Six features were computed:
    - commonWords: number of common words in the two utterances after deleting stop-words
    - firstLength: the length of the original utterance after deleting stop-words
    - secondLength: the length of the referred utterance after deleting stop-words

- question: Boolean feature which is true when question markers are present in at least one the utterances (*why, what, how, when, ?)*
- difference: Boolean feature which is true only if negation discourse markers are present in the original utterance (no, *not, cannot, must not, disagree)*
- agreement: Boolean feature which is true only if discourse markers specific to an agreement appear in the original utterance (*agree, true, ok, yes, confirm, right, exactly)*
- argumentation: Boolean feature which is true only if discourse markers specific to an argumentation/motivation are present in the original utterance (*so, because, for example)*
- class: nominal feature which can take one value in the set of the six categories already defined: {*adjacency, repetition, collaboration, convergence, argumentation, difference*}.

4. Computed features were written one on a line in the ARFF file.

Here's an example of the features computed for two utterances:

Original utterance:

and why is that, precisely?

Referenced utterance:

well, I am convinced that forums are the best way to aggregate global knowledge in a very seo friendly manner; hence easy to find as well

Table 8

Features generated for a pair of utterances

CommonWords	FirstLength	SecondLength	Question	Difference	Agreement	Pattern
0	5	26	true	false	false	adjacency

SMO Classifier

The SMO Classifier is an implementation of SVM (Support Vector Machines) algorithm developed by Vapnik in 1998, which is often used for binary classification. The algorithm tries to find a hyperplan to separate two classes with a maximal margin. SMO "implements the sequential minimal optimization algorithm for training a SVM classifier using Gaussian or polynomial kernels" [29]. Missing values are globally replaced and the nominal ones are transformed into binary ones after normalization. Normalization can be turned off or input data can be standardized to the mean value or to the unitary variance. For multiclass classification problems pairwise classification is used. This method trains the classifier for each pair of

classes, building $n=k*(k-1)/2$ independent binary classifiers, where k represents the number of classes. For associating a label to an input point, the prediction for each of the n classifiers is computed representing a vote. Most voted class is associated to the instance and for classes with the same number of votes a random decision is taken.

Results and conclusions

The training corpus used in the experiment consisted of three conversations summing up 442 utterances; the same corpus was used for testing and 10 folds cross-validation. Normalization was disabled during training. The overall classifier accuracy was 53.39%. Below are some statistics for the classifier and for each of the inter-animation patterns studied:

TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
0.655	0.414	0.61	0.655	0.632	0.65	adjacency
0	0	0	0	0	0.528	repetition
0.609	0.281	0.432	0.609	0.505	0.671	collaboration
0	0	0	0	0	0.734	convergence
0.392	0.043	0.541	0.392	0.455	0.757	difference
0.074	0.012	0.286	0.074	0.118	0.811	argumentation
Weighted Average						
0.534	0.285	0.496	0.534	0.505	0.674	

Fig. 1. Detailed accuracy for each class.

This experiment could not discover two of the inter-animation patterns described in the theoretical support: *repetition* and *convergence*. An explanation for that might be that these two patterns are harder to identify because the former is often mistaken with *adjacency pair* and the latter has not sufficient supporting examples in the training corpus and the features used to solve the general problem are not informative enough for this particular case.

a	*b*	*c*	*d*	*e*	*f*	←	*classified as*
144	0	67	0	6	3	|	*a = adjacency*
12	0	4	0	2	1	|	*b = repetition*
38	0	70	0	7	0	|	*c = collaboration*
6	0	3	0	0	1	|	*d = convergence*
19	0	12	0	20	0	|	*e = difference*
17	0	6	0	2	2	|	*f = argumentation*

Fig. 2. Confusion Matrix.

5. Comparison of the approaches

In order to have a clear picture about the performance of each of the three approaches used in the experiments, we must refer to the following aspects:

1. Speech Acts/Patterns considered: the first experiment deals with 20 labels for *Speech Act* column and with 10 labels for *Argumentation* column. The second experiment reduces the number of labels used for *Speech Act* column to 13, while the latter uses only 6 patterns.
2. Preprocessing: in the first experiment, default preprocessing included in TagHelper was used. This includes tokenizing, stemming, stop words removal and removal of rare features. The second and third experiments include partial or total punctuation removal and lemmatizing (only the second).
3. Classification algorithms used: Naïve Bayes (1st experiment), Maximum Entropy (2nd experiment) and Sequential Minimal Optimization algorithm based on SVM (3rd experiment)
4. The overall approach accuracy is presented in Table 9.

Table 9

Overall approach accuracy

TagHelper + Naïve Bayes	NLTK + MaxEnt	WEKA + SMO
~30%*	37.64%	53.39%

*TagHelper doesn't have a function to compute overall accuracy.
The number was computed taking into account *Speech Act* column.

Some conclusions can be drawn:

- accuracy increases as the number of labels considered decreases. This is a normal phenomenon because the first two experiments use too many labels and the difference in annotation between one to another can be made only by a human judge.
- punctuation is very important for the classification task (it was stripped off only in the third experiment, but was indirectly used for establishing the *question* characteristic of the utterance).
- SVM seems to solve better the classification task, among all the three experiments, thanks to numerical features introduced in the model: number of common words or the length of the two utterances.

The low values of the performances in the first two experiments may be explained by the limitations of the TagHelper system, which does not parse multiple annotations and by some annotation inconsistencies. In the third experiment, the accuracy suffers due to the usage of binary indicators (true/false) for some

features (question, difference, agreement, argumentation). The replacement of the values of these features with values form the interval [0..1] might help in a better classification of utterances expressing complex intentions (there are both negations and questions in the same utterance).

6. Conclusions

Three experiments were tried for determining speech acts, dialogue acts or links between chat utterances in CSCL chats. The first category associates actions to each utterance in a conversation, the second category extends this model by adding adjacency pairs and other notions specific to dialogues and the third is represented by logical, semantic or grammatical connections between two sentences/phrases uttered by different or the same user(s) of a chat. The tools used in the experiments are well-known NLP or machine learning tools – WEKA, NLTK and TagHelper – and the experiments involved mainly classification tasks. A heuristic approach was tried in the second experiment for determining links between utterances by using speech acts associated with those utterances.

The results of the experiments show that even simplistic approaches as Naïve Bayes classification and bag of words perform well for the given complex task, but better results were obtained with advanced classification techniques (SVM) and by choosing more informative features.

Improvements of the presented methods/algorithms include POS tagging, a better usage of WordNet for finding synonyms and antonyms and determining the topic of the conversations in use.

Acknowledgment

The research presented in this paper was partially performed under the FP7 EU STREP project LTfLL (http://www.ltfll-project.org/).

REFERENCES

[1] Anjewierden, A., Kolloffel, B., & Hulshof, C., *Towards educational data mining: Using data mining methods for automated chat analysis to understand and support inquiry learning processes*, paper presented at the International Workshop on Applying Data Mining in e-Learning, Crete, Greece, **2007**.

[2] Austin, J. L., *How to Do Things with Words*, Harvard University Press, **1962**.

[3] Bird, S., Klein, E., Loper E., *Natural Language Processing with Python*, O'Reilly, **2009**.

[4] Bunt, H., *Context and dialog control*, Think, Vol. 3, pp. 19-31, **1994**.

[5] Carenini, G., Ng, R.T., & Zhou, X., *Summarizing email conversations with clue words*, WWW '07, Proceedings of the 16th international conference on World Wide Web, pp. 91–100, **2007**.

[6] Collis, B., & Moonen, J., *Flexible learning in a digital world: Experiences and expectations*, London, Kogan, **2001**, **2002**.

[7] Fulks, H., & Pimentel, M., *Studying response-structure confusion in VMT*, G. Stahl (ed.), Studying Virtual Math Teams, Springer, **2008**.

[8] Hmelo-Silver, E.C., *Description of Chronologically-oriented Representation of Discourse and Tool Related Activity*, Proceedings of Workshop on Interaction Analysis at ICLS, Utrecht, **2008**.

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

[10] Jurafsky, D., & Martin H. J., *Speech and Language Processing*. *An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*, **2007**.

[11] Law, N., Lu, J., Leng, J., Yuen, J., Lai, M., *Understanding Knowledge Building from Multiple Perspectives*, Proceedings of Workshop on Interaction Analysis at ICLS, Utrecht, **2008**.

[12] Meijden, H., van der, *Samenwerkend leren in gameprojecten*, Electronic Version, Retrieved 17 January, 2007, downloaded from www.deonderwijsvernieuwingscooperatie.nl, **2007**.

[13] Mitchell, T. M., *Machine learning*, McGraw Hill, **2005**, **2010**.

[14] Orvis, K. L., & Lassiter, A. R. L., *Computer-supported collaborative learning: Best practices and principles for instructors*, Hershey, PA, Information Science Publishing **2007**.

[15] Rosé, C. P., Wang, Y.-C., Cui, Y., Arguello, J., Weinberger, A., Stegmann, K., et al., *Analyzing collaborative learning processes automatically: Exploiting the advances of computational linguistics in computer-supported collaborative learning*, International Journal of Computer Supported Collaborative Learning, 3(3), 237–271, **2008**.

[16] Sacks, H., *Lectures on conversation*, Oxford, UK: Blackwell, **1992**.

[17] Searle, J. R., *A taxonomy of illocutionary acts*, Gunderson, K. (Ed.), Language, Mind and Knowledge, Minnesota Studies in the Philosophy of Science, Vol. VII, pp. 344–369. University of Minnesota Press, Amsterdam. Also appears in John R. Searle, Expression and Meaning: Studies in the Theory of Speech Acts, Cambridge University Press, **1979**.

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

[19] Stahl, G., Koschmann, T., & Suthers, D., *Computer-supported collaborative learning: An historical perspective*, R. K. Sawyer (Ed.), Cambridge handbook of the learning sciences (pp. 409–426). Cambridge, UK: Cambridge University Press, **2006**.

[20] Stolcke, A., et al., *Dialogue Act Modeling for Automatic Tagging and Recognition of Conversational Speech*, Computational Linguistics, 26(3), 339-371, **2000**.

[21] Trausan-Matu, S., Chiru C., Bogdan, R., *Identificarea actelor de vorbire în dialogurile purtate pe chat*, **2004**.

[22] Trausan-Matu, S., Rebedea, T., *Inter-animation and polyphony in computer-supported collaborative learning*, Annals of the Academy of Romanian Scientists, Series on Science and Technology of Information, Vol. 3, No. 1, pp. 113-124, **2010**.

[23] Trausan-Matu, S., Rebedea, T., Dragan, A. & Alexandru, C, *Visualisation of learners' contributions in chat conversations*, J. Fong & P. Wang (Eds.), Blended Learning, pp. 215-224. Upper Saddle River, NJ: Prentice Hall, **2007**.

[24] Trausan-Matu, S., Stahl, G., & Sarmiento, J., *Supporting polyphonic collaborative learning.* E-service Journal, 6(1), 58-74, **2007**.

[25] Trausan-Matu, S., & Rebedea, T., *Polyphonic Inter-Animation of Voices in VMT*, G. Stahl (ed.), Studying Virtual Math Teams, Springer, **2010**.

[26] Trausan-Matu, S. *The polyphonic model of hybrid and collaborative learning*. In Wang, F.,L., Fong., J., Kwan, R.C., *Handbook of Research on Hybrid Learning Models: Advanced Tools, Technologies, and Applications*, Information Science Publishing, Hershey, New York, pp. 466-486, **2010**.

[27] Wang, Y., Joshi, M., Cohen, W., & Rose, C., *Recovering Implicit Thread Structure in Newsgroup Style Conversations*. ICWSM 2008: Proceedings of the 2nd International Conference on Weblogs and Social Media, pages 152--160, Seattle, WA, USA. Association for the Advancement of Artificial Intelligence, **2008**.

[28] Weinberger, A., & Fischer, F. *A framework to analyze argumentative knowledge construction in computer-supported collaborative learning*. Computers & Education, 46(1), 71–95, **2006**.

[29] Witten I., Eibe F., *Practical Machine Learning Tools and Techniques*. San Francisco, US: Morgan Kaufmann Publishers, **2005**.

[30] Wolf, F., & Gibson, E. (2005). *Representing discourse coherence: A corpus-based study*. Computational Linguistics 31:249–287, **2005**.

[31] Xi, W., Lind, J., & Brill, E., *Learning effective ranking functions for newsgroup search*, Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval, Sheffield, United Kingdom, July 25-29, **2004**.

[32] Yates, J., Orlikowski, W. J., & Woerner, S. L., *Conversational Coherence through Threading: Using Email Threads to Coordinate Distributed Work*, downloaded from http://seeit.mit.edu/Publications/ThreadingCoherence_27Nov06Working%20Paper.pdf, **2006**.

[33] Yeh, J., & Harnly, A., *Email thread reassembly using similarity matching*, Proc. Conference on Email and Anti-Spam (CEAS) '06., **2006**.