Assessing the Language of Chat for Teamwork Dialogue

Antonette Shibani[1, 3], Elizabeth Koh[1*], Vivian Lai[2] and Kyong Jin Shim[2]
[1]National Institute of Education, Nanyang Technological University, Singapore // [2]Singapore Management University, Singapore // [3]University of Technology Sydney, Australia // antonette.shibani@gmail.com // elizabeth.koh@nie.edu.sg // vivian.lai.2011@sis.smu.edu.sg // kjshim@smu.edu.sg
[*]Corresponding author

ABSTRACT

In technology enhanced language learning, many pedagogical activities involve students in online discussion such as synchronous chat, in order to help them practice their language skills. Besides developing the language competency of students, it is also crucial to nurture their teamwork competencies for today's global and complex environment. Language communication is an important glue of teamwork. In order to assess the language of chat for teamwork dimensions, several text mining methods are possible. However, difficulties arise such as pre-processing being a black box and classification approaches and algorithms being dependent on the context. To address these issues, the study will evaluate and explain pre-processing and classification methods used to analyze teamwork dialogue from a dataset of chat data. Analytics methods evaluated in this study provide a direction for assessing the language of chat for teamwork dialogue and can help extend the work of technology enhanced language learning to not only focus on academic competency, but on the communication aspect too.

Keywords

Teamwork, Pre-processing, Supervised machine learning, Text mining, Learning analytics

Introduction

In technology enhanced language learning, many pedagogical activities involve students in online discussion such as synchronous chat, in order to help them practice their language skills (e.g., Hedayati & Foomani, 2015). Such online team chat increases students' participation and evenly distributes their participation (Dennis & Garfield, 2003). In today's global and complex environment, besides developing the language competency of students, it is also important to nurture their teamwork and collaboration competencies (Rychen & Salganik, 2003). Moreover, language communication is an important glue in teamwork that helps build the team together and propel it forward in various tasks (Baker et al., 2005). Thus, beyond evaluating the academic competency of the language, it is meaningful to examine the teamwork discourse of students. Providing an assessment of teamwork dialogue helps students gain a better awareness of their teamwork competency and become better team players (Erkens & Janssen, 2008; Koh et al., 2014). Advancements in technology like learning analytics have allowed such assessments to be made more automated.

To develop such automated assessment systems, training models using human evaluation (involving computational linguistics and text classification) have the highest potential for efficiency and scalability. These models use computers to code the data instead of humans, which is faster; they can also be developed to examine different texts which allows for scalability (Erkens & Janssen, 2008; Rosé et al., 2008). In other words, language assessment can be made more efficient, and this facilitates more immediate feedback for students and teachers which can help improve students' learning (Anjewierden et al., 2007). However, a major challenge in automatically assessing online chat discourse is that chat texts have many irregularities in structure, short lengths and contextual complexities. This makes the identification of codes (such as teamwork dimensions) more difficult. Past research suggests that pre-processing the text to organize and take into account desired features would be helpful for analysis. Despite the importance of pre-processing, the steps in pre-processing tend to be vague. There seems to be a black box of pre-processing.

Another problem is that there are varying approaches to train models using human evaluation such as natural language processing and supervised machine learning. This area of analysis has not been widely documented and techniques also depend on the nature of the coding scheme and the purpose of classification. Furthermore, there are many machine learning algorithms that could be used. Finding out the most effective method would be the key to provide an automatic analysis of the language of chat, which serves as formative assessment for students.

Therefore, the focus of the paper is on the methods of using learning analytics for assessing online chat discourse, in particular, to measure the dimensions of teamwork. The research questions are:

- How can text be effectively pre-processed to assess the language of chat for teamwork dialogue?
- What approach works best to assess the language of chat for teamwork dialogue?
- Which algorithms are the most effective in classifying teamwork dimensions?

This study is part of a larger research project exploring the 21st century competency of teamwork in technology enhanced learning. Based on previous literature and pilot studies, six dimensions of teamwork were conceptualized (Koh et al., 2014). A coding scheme for these dimensions was developed and chat log data was manually coded. Chat log data of the study was obtained from 14 year old students who were collaborating on an online collaborative problem-solving activity as part of their project work curriculum. Besides providing students with the opportunity to engage in teamwork, the activity was designed for English communication, which also helps students practice their language skills. Baker et al. (2005) conceptualizes that communication "is the glue that holds the team together" (p. 240). In that sense, while there are specific teamwork dimensions, communication is an important part of the team as it joins the team together. In other words, while we measure teamwork dimensions, we are also indirectly measuring their language communication competency. Thus, it is possible to infer that if students have high teamwork competency dimensions, their communication skills will be high.

This paper will contribute to the research on learning analytics methodology for language learning. It attempts to open up the black box of pre-processing, revealing hidden steps in many learning analytic studies. The paper will also reveal approaches and algorithms most effective in identifying teamwork dimensions, which will be relevant to educational data mining and learning analytics researchers. Lastly, the empirical implementation and results provide an applied context for using learning analytics, and offer some insights for educational practitioners in the potential uses of learning analytics.

The paper begins with a brief literature review of the methodology involved in text mining in online chat. This is followed by further details of the background of the project and the dataset. Next, we elaborate on the methods of the study and look at both pre-processing and classification approaches. Subsequently, we test out our methods and discuss the results. The conclusion section deliberates on the practical and theoretical implications of the study.

Literature review

Online collaboration platforms such as Moodle, HipChat and Slack provide insights into individual and organization-level communication and collaboration behavior (Anders, 2016), help identify indicators of academic performance in students' online forum participation (Romero et al., 2013); they also identify patterns of student interactions and participation (He, 2013). Moreover, automated text mining analysis allows students to make better sense of the learning process during their collaboration, and possibly provide support and remediation (Anjewierden et al., 2007; He, 2013). This is an application of "Learning Analytics" which is defined as the "measurement, collection, analysis and reporting of data about learners and their contexts, for purposes of understanding and optimizing learning and the environments in which it occurs" (Siemens & Gašević, 2012, p.1).

Text mining traditionally analyzes long text such as essays and newspaper articles. However, chat text length is short and has irregularities in structure and contextual complexities. One of the key decisions for learning analytics researchers is whether to pre-process such text. Pre-processing is a core task in text mining to clean up the raw data and structure it for analysis. In structured text, the general pre-processing steps include: removing punctuations, trimming white spaces, performing stemming and POS tagging, and removing stop-words. Pre-processing can help clarify the meaning of text, but is also highly dependent on the data and the context. Steps of pre-processing are not widely documented in learning analytics. Many studies only do a basic pre-processing of text data since the usefulness of contextual pre-processing has not been studied in depth. Some studies choose not to perform pre-processing, as the original features of the text, e.g., punctuations, can be useful to derive meaning (Villatoro-Tello et al., 2012). On the other hand, a study on automatic topic detection in chats used sessionalization and extraction of features from icon text and URLs for pre-processing (Dong et al., 2006). As aforementioned, chat texts are rather different from conventional text, and could require more or different ways of pre-processing.

Several systems have been developed to take care of these issues in chat text pre-processing using the process of normalization (Clark & Araki, 2013; Han et al., 2012; Rosé et al., 2008). In addition to the misspelled words, normalization also corrects common chat language comprising out of vocabulary words (OOV) such as acronyms, abbreviations and emoticons. These systems are either dictionary-based or algorithm-based. A dictionary based approach was shown to outperform sophisticated algorithm approaches in terms of F score and word error rate by generating possible normalization pairs (Han et al., 2012). In a system called Casual English Conversion System (CECS), eight categories of irregularities such as Shortform (abbreviation), Shortform

(acronym), Typing error/ misspelling, Punctuation error/ omission, Non-dictionary slang, Cultural reference/ in-group meme, Wordplay/ intentional misspelling, and Omission of vocabulary were captured (Clark & Araki, 2013). This system preserves the phrase structure in texts and makes it more context-aware, hence useful for our work.

Pre-processing is typically followed by classification, where a discrete category is predicted based on the training data. Most approaches use human coding as the standard to check if the automated coding classifies instances similar to the standard (Anjewierden et al., 2007; Erkens & Janssen, 2008; Rosé et al., 2008). One application of classification approaches is in the identification of "topic" when there are multiple participants contributing text into a single common chat room.

There are several classification approaches depending on the nature of the coding scheme and the purpose of classification. One such approach is a rule-based classification based on the textual features and patterns identified for each dimension (Aggarwal & Zhai, 2012). Such classifiers are written using Natural Language Processing (NLP) rules for automating qualitative data analysis. Crowston et al. (2012) used a symbolic approach by looking for evidence of specific behavioral patterns and writing rules for automating content analysis. Different levels of analysis from the lowest phonological to the highest pragmatic linguistic approaches can be done. They do not require large datasets like statistical approaches, but are not easily transportable to other domains. The results of automated coding were compared with manual coding and a good performance on a number of codes was seen in terms of precision and recall measures. The study prioritized higher recall over precision, so that it can act as a support system for human coders to revisit and change codes if necessary. In another study, Erkens and Janssen (2008) developed an automatic coding procedure for 29 dialogue acts using the Multiple Episode Protocol Analysis (MEPA) computer program consisting of if-then rules using discourse markers and cue phrases for pattern matching. Another approach is supervised machine learning, where the aim is to classify messages according to a pre-determined set of categories. Algorithms such as Naïve Bayes, Support Vector Machines and k-Nearest Neighbor are used in machine learning to automatically categorize chat messages (Anjewierden et al., 2007; Rosé et al., 2008) in an educational setting to analyze student chat conversations. In line with the aforementioned studies, the current study will use and evaluate pre-processing and classification techniques to identify teamwork dimensions from students' chat data.

Research context, corpus and coding scheme

This research is part of a larger study on teamwork. A teamwork competency awareness program was designed for 14 year old students in a Singapore secondary school, for them to gain an awareness of their teamwork competency. During one section of the program, student teams of 3 or 4 used an online group chat to complete a problem-solving activity. The first task was an ice-breaker activity, while the second was a dilemma task. Students had about 45 minutes to complete the tasks using computers in school during their class time. Students were seated away from their team members to reduce face-to-face communication and encourage online communication. Instructions were provided to the students through the chat in real time by the Chat Administrator, who is a researcher. As one of the aims of the study, we hoped to help students gain an awareness of their teamwork competency through multiple measures. One measure was the assessment of teamwork dimensions of the online chat dialogue.

The six dimensions of teamwork competency measured are: coordination (COD) - organizing team activities to complete a task on time; mutual performance monitoring (MPM) - tracking the performance of team members; team decision making (TDM) - integrating information, selecting the best solution, and evaluating the consequences in a team; constructive conflict (CSC) - dealing with differences in interpretation between team members through discussion and clarification; team emotional support (TES) - supporting team members emotionally and psychologically; and, team commitment (TCM) - identifying with and being involved in team goals (See Koh et al., 2014 and Koh et al., 2016 for further descriptions). A sample of the coding scheme is provided in the Appendix.

A total of 272 students participated in the online activity in 76 teams. This resulted in 19762 raw chat messages, inclusive of spam lines. This formed the corpus of the study. Two coders annotated 7 teams' data with a Cohen's Kappa > 0.65. They then proceeded to code the rest of the teams individually. For this analysis, data from 34 teams coded manually was taken for automated analysis. A sub-section of the dataset was selected for this analysis due to practical reasons as the approach of NLP rules requires substantial time for rule creation. This dataset had 9783 lines in total which was split into test and training sets. The minimum was 71, the maximum was 487, and the average was 287 lines per team. The training set consisted of 5705 lines and the test set, 4078

lines. The training set was selected using maximum variation sampling to contain diverse data based on participation, class and team composition. It represented data from high and low participating teams in terms of chat lines, teams from all 7 classes and both 3 and 4 member teams, for the results to be more generalizable to new data.

Methods and methodology

The different approaches and methods developed are described in this section.

Pre-processing

With regard to text pre-processing, two types of text were created: (1) non-pre-processed text, termed "base," and (2) pre-processed text.

For the **base text**, the steps comprised:

- Removing unicode text from raw text - to overcome Python unicode decode error
- Situation coding and spam filter - this was a manual coding performed to categorize messages into situations and spam/no code (Shibani et al., 2015).

The chat lines were grouped by topics for eight situations starting from Introduction (ST1) to Team Dismissal (ST8). These situations reduce the ambiguity in the context of data and help in writing rules for classification. The messages that need not be coded for teamwork dimensions (mostly spam) were also coded as "no code" (nc) manually while performing situation coding. With our reasonable amount of data, it was easier to do manual spam detection since developing a context-sensitive spam detection system was out of scope.

While some of the spam lines seem to not interfere with the text classification, some do, since they contain keywords from teamwork dimensions. Certain dimensions like COD and TES follow the same structure and hence it becomes necessary to distinguish the chat lines from nc lines that need not be coded. Also, these lines when given as input decrease the performance of machine learning algorithms, since the machine learning systems will extract features from this text and also predict categories for them. The rules defined to isolate nc lines can be found in the Appendix.

For the **pre-processed text**, the main idea was to simplify the text for the classifier to learn the features easily. For example, there can be just one feature {{Name}} in a feature set, instead of having the classifier to learn 100 names from the text as features. We hope that this will help in grouping similar features together to build a better predicting classifier, which is also more memory and time efficient.

The steps for base were similarly performed and additional pre-processing techniques were carried out. This extended pre-processing mechanism covers several features of chat data and preserves useful nuances in text. Figure 1 shows the different steps of pre-processing that were performed for normalizing chat data. We next describe each of these steps.

Emoticons and punctuation tagging

As emoticons and several punctuation marks were important to our coding scheme, we converted them to indicators (tags). A list of emoticons were compiled from online sources and our own data, and grouped into categories (See Appendix). Other punctuation marks that were not relevant to the coding scheme were removed.

Chat abbreviation expansion

Chat language contains many abbreviations in the form of short forms and acronyms that have to be expanded. Short forms are shorter representations of a word by omitting or replacing few characters e.g., grp → group. Acronyms are formed from the first character of each word e.g., lol → laughing out loud. These words may be omitted by keyword search due to incorrect match if not expanded. We created a dictionary of these words from urban dictionaries (see http://www.urbandictionary.com/) and acronym dictionaries (see http://www.singlishdictionary.com/) to replace abbreviations by expansions. Relevant words were also added

from the CECS database for more comprehensive coverage for other corpus data (Clark et al., 2013). It is to be noted that all chat abbreviations cannot be replaced by an expansion, as some meanings vary in different contexts, e.g., "btw" may refer to "between" or "by the way." Contractions and expansions in English are also taken care of in this step, e.g., can't → cannot. Words specific to our context were also added as they occurred frequently in the chat, e.g., ans → answer.

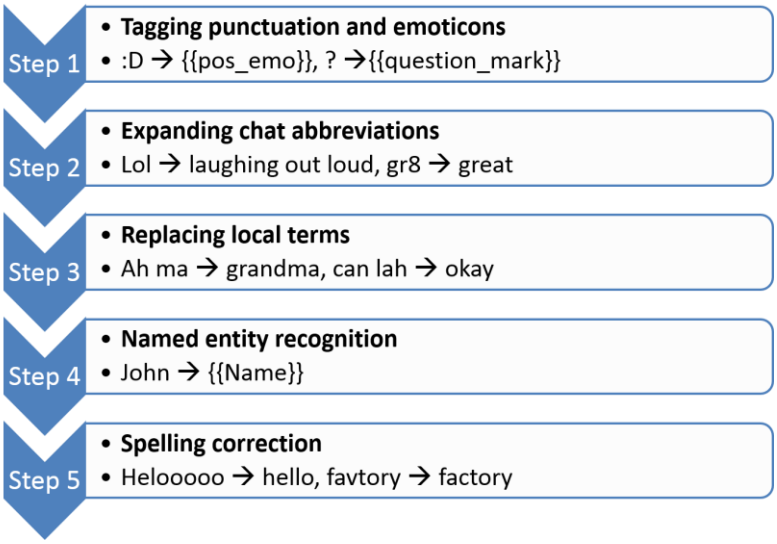

Figure 1. Steps in pre-processing chat data

Local terms replacement

In addition to the chat terms, there were also local terms found in the corpus, whose meanings cannot be found in an English dictionary. These Singlish terms derived from the local languages are extensively used in chat by students. We constructed a local dictionary with Singlish terms and their English equivalents for use in our system. The words in this dictionary were gathered from websites which provide meanings for Singlish terms and also from our own corpus. The local terms were replaced with their equivalent English terms using this dictionary, e.g., ah ma → grandma, cher → teacher, kampung → home.

Named entity recognition

The reference of names could not be ignored in our context as it indicates teamwork dimensions in different circumstances according to our coding scheme e.g., in COD for discussing who is in the team ("John's in our team"). Named entity recognition identifies names and labels the different elements of text into predefined categories like person, organization, numbers etc.

We used the Stanford NER caseless class model trained for CoNLL and MUC data that tags three entities: Person, Location, Organization, Misc (Finkel et al., 2005). The NER is part of the Stanford CoreNLP that contains a list of open source tools for natural language analysis. The caseless model that ignores capitalization was selected as most of our chat data did not follow proper capitalization of names. We ran a manual check to remove the wrongly identified Person tags. The pre-processing script then tagged the names found in the chat as {{Name}}. Training our own classifier for local names could be part of future work using more data, as the existing Stanford NER models cannot be extended.

Spelling correction

Spell checking is performed as the next step of the pre-processing pipeline on all word tokens excluding the already tagged ones from the previous steps. Upon this transformation, all words were tagged with a {{S}} prefix to indicate that they were changed. The engine is written in Python using PyEnchant package (see https://pypi.python.org/pypi/pyenchant/) and works in three stages as shown in Figure 2.

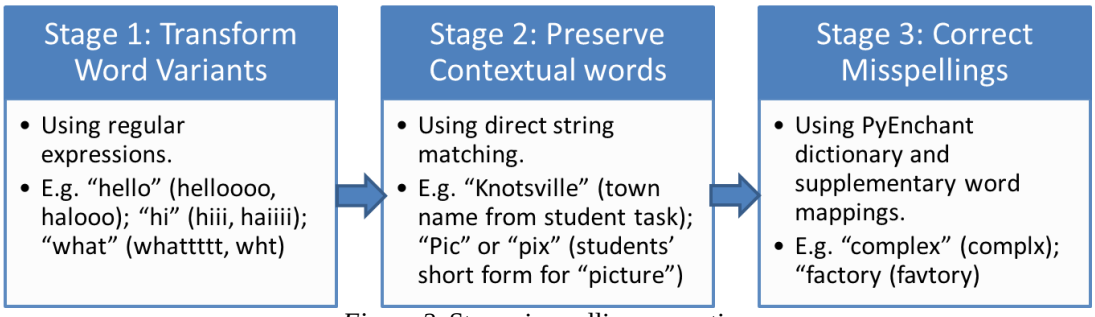

Figure 2. Stages in spelling correction

Classification

The goal of our classification task was to classify chat lines into zero, one or more teamwork dimensions. The main issue for automatic classification is that we require a single line to be coded for multiple dimensions (multi-label classification), whereas most algorithms classify only one category to an instance (binomial or multi-class classification). In multi-class classification, even if there are six classes of teamwork dimensions, they are mutually exclusive and each chat line can be categorized as only one of the dimensions. In multi-label classification, each instance can be assigned multiple labels, like a chat line indicating both COD and TES can be assigned both. In our current implementation, we have classified each dimension separately as 0 or 1 (binomial classification) for a line and then combined all dimensions together to get the final predicted output.

The two approaches below were implemented for classification:

- NLP rule-based classification
- Supervised machine learning

We shall discuss the implementation of each method in detail and then evaluate their performances using our dataset.

NLP rule-based classification

Our first approach was to do a rule-based classification based on the textual features and patterns we identified for each dimension (Aggarwal & Zhai, 2012). The unit of analysis is a chat line. Text is represented as a string with a sequence of words in order to preserve useful information like the sentence patterns and phrase structures in the chat lines. To maintain the contextual link between the preceding line and the next line and sentence cohesion, the order of chat lines was also preserved. A search is performed on the identified features in the text and the resulting matching text are coded for teamwork dimensions. This system was developed in R. The rules are specific to each dimension, but the predicted labels are combined to obtain the final coded output. Table 1 displays a sample.

The indicative terms dictionary consists of keywords that are indicative of teamwork dimensions and forms the basis of the classification system. The dictionary may contain individual words or phrases depending on the context. The following steps are used to write rules based on the indicative terms (Shibani et al., 2015).

Existence: Some keywords from the indicative terms dictionary are strongly indicative of the teamwork dimensions. This means that the occurrence of such words or phrases demonstrates that the chat line contains a particular teamwork dimension. For example, a positive emoticon will always indicate TES and hence it can be detected by its existence. Similarly task-specific words like factory, pollution etc. indicate TDM. For each dimension, an input string is searched for such terms from the indicative terms dictionary. If the term is present, the teamwork dimension is coded 1, else it is coded 0.

Frequency: The frequency of indicative terms can indicate the strength of a dimension in a chat occurrence. We do not measure the strength of a dimension in the current implementation and we only measure the presence, i.e., a line containing 5 keywords for TDM and another line containing a single keyword for TDM are both coded the same 1 for the dimension. In systems that measure this variation, such rules for frequency can be written. However, we used the n-gram tokenizer from Weka (see http://weka.sourceforge.net/doc.dev/weka/core/tokenizers/NGramTokenizer.html) to find the frequent bigrams

and trigrams in addition to single words that occur in the chat lines and determined if they contributed to teamwork dimensions. These frequent words identified from the chat text were added to the indicative terms dictionary rather than manually looking for keywords, thus reducing human effort.

Table 1. Sample coding for teamwork dimensions in a chat log

Name	Message	Situation	COD	MPM	TDM	CSC	TES	TCM
A	Hellooooo	ST1	0	0	0	0	1	0
D	Hi :D	ST1	0	0	0	0	1	0
A	WHO ELSE AH?	ST1	1	0	0	0	0	0
B	LET S DO THIS	ST2	1	0	0	0	0	1
Admin	a) Describe your ideal teacher b) As a team, decide on the three most important qualities of an ideal teacher	ST2	0	0	0	0	0	0
B	DESCRIBE YOUR IDEAL TEACHER :|	ST2	0	0	1	0	0	0
A	bob describe cher sia	ST3	0	1	1	1	0	0
B	understanding lor	ST3	0	0	1	1	0	0
C	DISCRIBE CHER????	ST3	0	0	1	1	0	0
B	a) understanding b) respectful ,dont give much homework	ST4	1	0	1	1	0	0
A	okay done	ST4	1	0	0	0	1	0
B	:like:	ST4	0	0	0	0	1	0

Proximity: Proximity can be used to identify the context of words in utterances. For example a salutation followed by name can be identified as a teacher's name rather than a student's name ("Mr Wong is cool, agree Amy?"). Rules based on the proximity of keywords were identified from the corpus. While the rules involving keywords within the same utterance in a particular proximity are relatively easier to implement, the rules involving threaded utterances are more complex. An example is finding a word corrected for spelling by a student from the previous line (Line 56: Write about your idel teacher, Line 58:*ideal). This involves bringing the previous and next lines into context and requires complex implementation.

Weightage: There are a few keywords that indicate one dimension but also occur in other contexts. Such words were removed or added based on the error analysis of confusion matrix, obtained by comparing human and automated coding, giving weightage for one dimension over another. We aimed to reduce false positives where the line is coded for a dimension when it is not indicative of that dimension (0-1, where 0 denotes manual coding and 1 denotes automated coding). For example, if a keyword for TDM increased the 0-1 pairs for COD with no considerable increase in 1-1 pairs of TDM, it was removed.

Complex patterns: More complex patterns involving coded situations are useful in identifying the context of text. For example, task related keywords should occur in the designated situations like ST3 and ST6 where the task discussion occurs, and greetings occur in ST1. A name occurring in ST1 is part of a rule for COD, hence a related rule can be written to distinguish it from MPM (ST1 + name in text represents COD). Also, typo correction by self or other members can be used to distinguish MPM or TES, with the help of complex rules.

Although the above approach looked fine for the data we analyzed after initial coding, there are inherent disadvantages in this method. The key issues are as follows:

Generalizability to new data: Unless the future data is very similar to the current data, the classifier will not generalize, which is the case in most real-world scenarios. The students from a different school can have a completely different style of writing. This leads to the case where the teamwork dimensions are not captured by the classifier, because the words used are different from the previously identified words and patterns that the classifier was trained and tested on. An ideal classifier should be able to generalize to new data by learning features that can be applied to new data without high variance.

Labor intensive rule writing: The process of identifying rules from the data and implementing them in a classifier is a time consuming and laborious process. If this process has to be repeated for all the new data, it increases the workload. Also, if there are changes in the coding scheme, the rules may have to be modified to fit the new coding scheme. Such manual feature extraction process makes the effort demanding and strenuous.

Context-sensitive rules: The NLP rules which are used are human-context sensitive and the rules can be written in different ways by different researchers, giving inconsistent results. It is very difficult for a researcher to reproduce/replicate another researcher's rules just by looking at the instructions and data, since they may identify different patterns. Also, there is no common standard for rules or guidelines for the number of rules. Another related issue would be overfitting, when too many rules are added such that it reduces the classifier's ability to generalize.

Supervised machine learning

Our second approach was to implement a classifier using machine learning algorithms. Since the human coded labels were available for training the classifier, supervised machine learning was used. Supervised learning algorithms learn features from the labelled instances and use them to predict the category for new unlabeled instances based on the likelihood suggested by a training set (Kotsiantis et al., 2007). The same training and test datasets from NLP rules classifier were used for analysis so that both the methods can be compared.

The advantage of this method is that the classifier can extract features from the training data automatically, so there is minimal human effort in identifying features. It also makes use of existing implementations of the different machine learning algorithms, so there is no need for a new development of a system except for tuning the required parameters to obtain optimal classifiers.

The scikit-learn Python package (see http://scikit-learn.org/stable/) was used for this implementation as it allowed customizable scripting and the use of different algorithmic implementations without requiring expert technicalities. We implemented the following six commonly used algorithms for our analysis and comparison (Aggarwal & Zhai, 2012, 2010, Kotsiantis et al., 2007):

- Decision Tree (DT)
- K-Nearest Neighbors (KN)
- Logistic Regression (LR)
- Naive Bayes (NB)
- Single-layer Perceptron (PE)
- Support Vector Machine (SV)

Our machine learning classifiers used the textual chat message as input to extract features. Both the base text and the pre-processed text of the messages were used as inputs for comparison. The spam and "nc" messages were removed from the training and test data so that the classifier learns better features as explained in the earlier section. To represent the text document as a vector for machine learning, the documents are typically stored in terms of the word features and its occurrences in the document. For our data, the messages were vectorized based on counts. In large corpus, the tf-idf would have worked better since it captures the words which are more interesting in the given documents by considering both the term frequency and the inverse document frequency. The CountVectorizer from scikit-learn was used to tokenize and extract both unigrams and bigrams. Using both unigrams and bigrams as features gave better results than using unigrams only. This was probably because our data had useful features that occurred together as bigrams. Feature selection based on statistical tests did not improve the classification results since it removed useful features from our small dataset, thereby decreasing performance, so we have currently used all the extracted features.

Future work will focus on implementing a multi-label algorithm, where all six dimensions can be classified by the same classifier rather than combining multiple binary classifiers. This will overcome the problem of finding dependencies among the dimensions. The disadvantage of this traditional bag-of-words approach is that it does not consider the semantic relations between words, so it is difficult to improve the accuracy of these classification algorithms unless more features are identified from a bigger dataset.

Results and discussion

To address research questions 1 and 2, four common metrics were calculated: Cohen's Kappa, F score (sometimes called F measure), precision and recall (Powers, 2011). Accuracy was not calculated due to certain limitations (e.g., skewed classes with unbalanced data). Human coded data was considered the gold standard, and compared with machine coded data. Cohen's Kappa (see http://www.pmean.com/definitions/kappa.htm) is a commonly used statistical measure to calculate inter-rater reliability between two coders, controlling agreement by chance. Minimum acceptable Cohen's Kappa values range from 0.4-0.6. Precision is defined as the ratio of

the number of correctly coded cases by the machine learning model to the total number of cases coded by the model. Recall is defined as the ratio of the number of correctly coded cases by the model to the total number of correctly coded cases specified by standard human coding. A model has high precision when most of its cases are coded accurately, which however, reduces recall since it does not code many cases. To maintain a balance between these two measures, the F score calculates the harmonic mean of the precision and recall to create a single measure for model effectiveness. Although the preferred values for a good model depends on the context of data, 0.6 and above is considered an acceptable standard for reporting these measures in educational data mining.

These metrics were calculated for each of the six teamwork competency dimensions. A comparison between the type of text, namely base and pre-processed text was performed for each of the two classification approaches, NLP rule-based classification (NLP rules) and supervised machine learning (Sup ML). Table 2 reports the performance metrics.

Table 2. Results of using NLP rules and supervised machine learning on base and pre-processed text*

	COD		MPM		TDM		CSC		TES		TCM	
	NLP rules	Sup ML	NLP rules	Sup ML	NLP rules	Sup ML	NLP rules	Sup ML	NLP rules	Sup ML	NLP rules	Sup ML
Base												
Cohen's Kappa	0.35	*0.53 (SV)*	0.18	*0.36 (NB)*	0.69	*0.72 (SV)*	0.56	*0.57 (SV)*	0.28	*0.71 (SV)*	0.28	*0.94 (SV)*
F score	0.46	*0.63 (DT)*	0.21	*0.42 (NB)*	0.80	*0.83 (NB)*	0.69	0.67 (NB)	0.37	*0.78 (SV)*	0.37	*0.95 (SV)*
Precision	0.59	*0.81 (LR)*	0.73	*0.83 (LR)*	0.85	*0.92 (KN)*	0.58	*0.79 (LR)*	0.82	*0.93 (LR)*	0.82	*1 (KN)*
Recall	0.38	*0.62 (DT)*	0.12	*0.33 (NB)*	0.75	*0.83 (NB)*	0.84	0.78 (PE)	0.24	*0.88 (DT)*	0.24	*0.93 (DT)*
Pre-processed												
Cohen's Kappa	0.29	***0.56 (SV)***	**0.29**	***0.59 (SV)***	0.69	***0.74 (NB)***	0.55	***0.60 (SV)***	**0.75**	***0.83 (SV)***	**0.75**	***0.95 (SV)***
F score	0.46	***0.64 (SV)***	0.33	***0.64 (SV)***	0.80	***0.84 (NB)***	0.68	***0.69 (NB)***	**0.81**	***0.88 (SV)***	0.81	*0.95 (SV)*
Precision	0.43	***0.79 (LR)***	**0.78**	**0.78 (LR)**	0.84	***0.96 (KN)***	0.56	*0.77 (LR)*	**0.91**	***0.95 (LR)***	**0.91**	*1 (KN)*
Recall	**0.51**	*0.62 (DT)*	**0.21**	***0.59 (NB)***	**0.77**	***0.84 (NB)***	**0.85**	**0.79 (NB)**	**0.73**	***0.89 (DT)***	**0.73**	***0.94 (DT)***

Note: Results for the highest performing classifier algorithms are recorded, with the algorithm listed in the brackets. In bold are the metrics in which pre-processed text was larger than base text. In italics are the metrics in which Sup ML was larger than NLP rules.

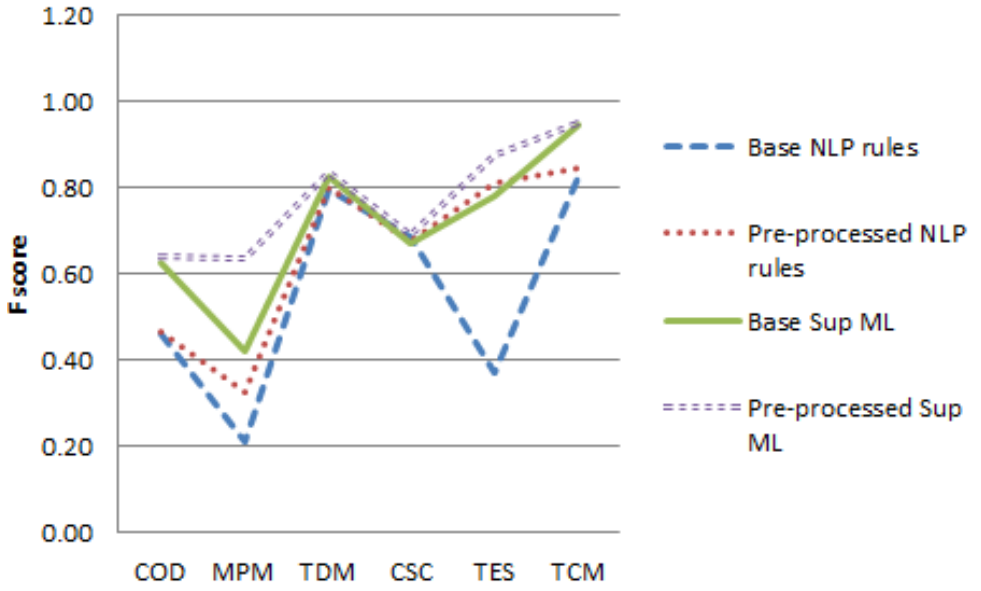

Figure 3. F score performance of text type and classification approach for the 6 teamwork dimensions

Figure 3 provides a graphical representation of the F score for two comparisons of text type and classification approach. As can be seen, pre-processed text performs better than the base text on most teamwork dimensions. Examining F score results of NLP rules, pre-processed text outperformed base text in 3 out of the 6 dimensions, tying in 2 dimensions, and decreasing by 0.01 for CSC. More obviously, F score results of Sup ML showed better performance of pre-processed text compared to base text for 5 of the 6 dimensions, and tying with one dimension, TCM. This suggests that pre-processing text is better than non-pre-processed text with regard to our dataset and features.

As for classification approach, it is clear from Table 2 that Sup ML outperforms NLP rules in most of the metrics. The only exception was in CSC where the NLP rules were written to mimic human coding by using TDM as a rule, since a part of TDM lines also fall under CSC. In machine learning, since the dimensions were independent, there was no link between TDM and CSC. Since the classification of CSC was only based on the word features in machine learning, it performed lower. This suggests that Sup ML is a better method for the dataset and features compared to NLP rules in general. It is also better to go with Sup ML for scalability reasons, as it is able to learn features for new and larger datasets. However, one limitation of our current manual coding scheme is that it takes into account not only words, but turn-taking and member roles. Also, there is need for a larger dataset as well as other datasets to test out the performance of the approach. Further work is required to refine the machine learning features too. Nevertheless, NLP rules are a good second choice as these rules make sense to the users and are rather intuitive, although the rules are not too scalable for other contexts and tasks.

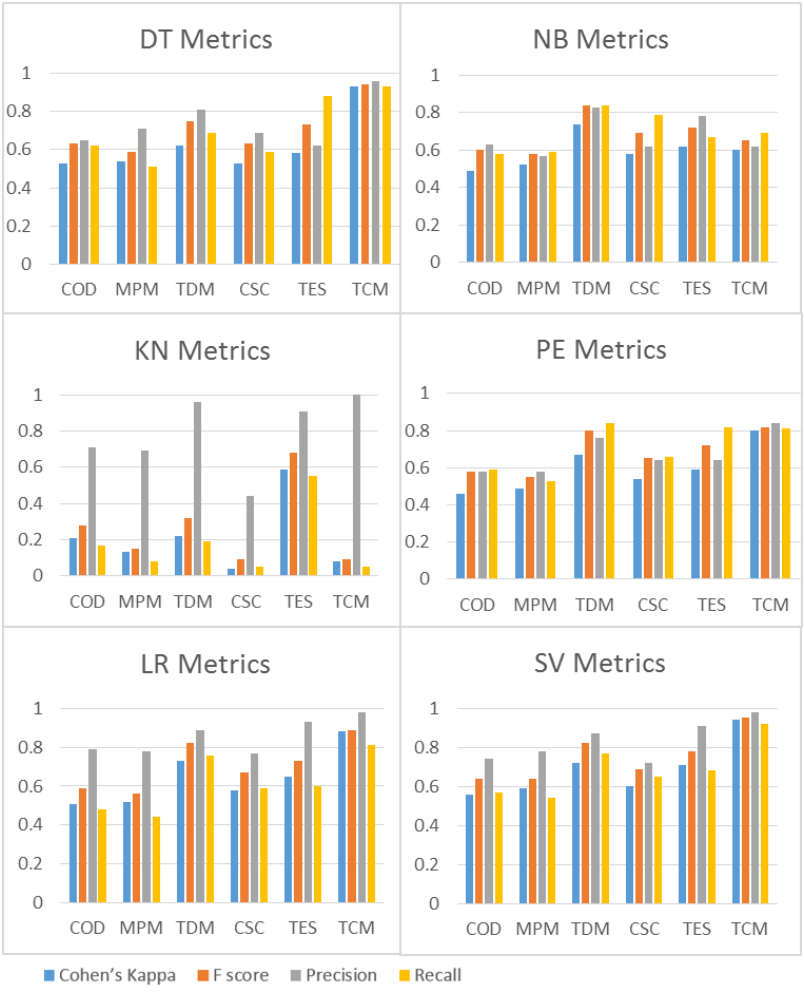

Figure 4. Performance metrics of all algorithms

Next, we examine research question 3, which compares how the 6 different algorithms classify teamwork dimensions. For the 6 dimensions, it was possible that different classifiers performed better in different dimensions, which was also observed in our analysis. There was no single classifier that performed the best among all dimensions, though SV performed higher than others in general. Figure 4 reports the results of all the metrics of the 6 algorithms. As can be seen, SV has the highest F score among the other 5 algorithms on almost

all dimensions of teamwork. It has also hit at least a 0.6 performance benchmark. Although SV has lower precision, recall is higher. This suggests that SV predicts more instances giving more coverage of dimensions implying that it is more complete than exact. The SV edges the other algorithms especially in terms of MPM and TES. It can also be seen that KN is the least effective algorithm for our coding scheme and dataset. These results are similar to Altrabsheh et al. (2014) who found that the SV algorithm had the highest F score, precision and recall for learning sentiment from students' feedback.

KN is often seen to be intolerant to noise in the data which means that the irrelevant features in text corrupt its output. The irrelevant features in our dataset could have been an important reason for KN to perform badly for our data. SV on the other hand, removes irrelevant features and performs better. SV is suited to tasks with relatively larger number of features than number of instances since it only selects fewer support vectors. PE is also seen to have an advantage when there are few relevant features among many features due to their superior time complexity. One reason for NB to work reasonably well in our dataset is that it only requires a small dataset to train the classifier by estimating its parameters. It classifies correctly when the category is more probable than others. The classifier also ignores rough estimates in its underlying probability model to give a good overall classifier.

Other factors like runtime and memory were not considered for performance measures since the feature set was small and hence the training and test times were negligible. We intend to use the identified teamwork dimensions from this method by converting the occurrences into a scale, to provide an aggregated measure so that users (such as students) know the degree to which they have displayed the different dimensions of teamwork. Thus, measuring these teamwork dimensions serve as a formative assessment of students' competencies.

Implications and conclusion

In this paper, we have evaluated and explained the pre-processing and classification methods used for analyzing teamwork from chat data. The study has found that pre-processed text is better than base text and re-iterates the importance of pre-processing according to the context of application. The pre-processing system has uncovered the complexities in chat language to better prepare the data for classification. It reveals unconventional English patterns in the educational context which are not covered in the existing corpuses. The process of how to deal with these irregularities is explained in detail which opens up the black box of pre-processing. This workflow for pre-processing chat data is a novel contribution, which can be potentially applied to other free flowing online chat text for normalizing raw chat to meaningful data. While time-consuming, our study suggests that developing and implementing the pre-processing techniques is a worthwhile investment.

The two different methods for classification have also been investigated using our dataset. The machine learning method seems to outperform the NLP rules for classification in terms of performance results. Practical considerations such as time and effort are reduced in machine learning with better scope for scalability. In supervised machine learning, we have also explored the capabilities of the different learning algorithms by comparing their performances. In general, the Support Vector Machine algorithm produces better performance for most teamwork dimensions in our dataset. This aligns with several papers in text mining of the effectiveness of the SV algorithm (Altrabsheh et al., 2014; Dong et al., 2006).

While the focus of our paper was to explain different methods to analyze online chat discourse with respect to teamwork dimensions, it should be noted that the results are specific to our data and the methods need to be selected appropriately for any given problem according to the context. Many of our rules also need further refinement. As shown, the metrics of the approaches can be further improved. Work is in progress to increase the performance of the classification approach. This will allow the study to reliably and relatively accurately categorize teamwork dimensions in a chat log. Future work can also include more quantitative "use frequency data" for analysis such as trace logs and counts of participation to see if they contribute to dimensions of teamwork. In the current context, such counts were not included due to spam lines in the data and the focus was more on assessing the quality of chat text.

Our current contribution enables efficient measurement of 6 teamwork dimensions in a semi-automated manner which can be used in a broad range of language learning situations. These dimensions do not require specific team member expertise, which is ideal in many foreign language learning situations where homogenous teams of students with no particular expertise in the language are learning together. This is a first step towards generating an automatic assessment of the chat language.

Moreover, this approach maximizes the learning experience of students. The learning process of students in online chat is used more than just for communication and topical dialogues, but the process itself is analyzed and becomes an artefact and means of learning. It becomes a reflective tool for students and part of their formative assessment. The research hopes to visualize the results of the automatic assessment in a learning profile that helps students better understand their language use and behaviors. We envision a scenario where this learning profile is shown at key points in the course so that students can stop, reflect and be more aware of their discourse. This feedback will go towards helping students understand their dialogic strengths and weaknesses for further improvement.

This approach also supports teachers and could make them more effective in large classes. As the system is easily scalable, in bigger classes, chat language can be automatically assessed and feedback provided efficiently to the students. The teacher is then able to zoom in on students who are not performing well, and intervene accordingly. For instance, a teacher can group large class sizes into smaller groups of students and provide them with a chat topic. After students complete their chat activity, the system can measure the teamwork dimensions and subsequently display the dimension scores to students. This provides feedback and allows students to become more aware of their teamwork competency dimensions. Teachers can also view the dimension scores of students and speak to the students who do not perform well. Overall, this reduces the load of teachers from assessing, and allows them to focus more on helping students to learn. That said, the teachers' role in intervening in a meaningful manner is still crucial. While this method and approach makes certain communication skills of students visible, teachers still need to scaffold and help students make sense of the information and suggest ways for improvement.

The learning analytics methods evaluated in this study provide a direction for assessing the language of chat for teamwork dialogue. The approach and system will be useful for both students and teachers in evaluating the students' communication ability. It will help extend the work of technology enhanced language learning to not focus only on academic competency, but the communication aspect too as informed by teamwork dialogue. Ultimately, the study provides a means of assessing chat dialogue such that students are able to gain a better awareness of their teamwork competency which is an important end in itself.

Acknowledgements

This paper refers to data and analysis from the research projects OER62/12EK and OER09/15 EK, funded by the Education Research Funding Programme, National Institute of Education, Nanyang Technological University, Singapore. The first author has subsequently moved to another institution. The views expressed in this paper are the authors' and do not necessarily represent the views of NIE.

References

Anders, A. (2016). Team communication platforms and emergent social collaboration practices. *International Journal of Business Communication, 53*(2), 224-261.

Aggarwal, C. C., & Zhai, C. (2012). A Survey of text classification algorithms. In *Mining text data* (pp. 163-222). doi:10.1007/978-1-4614-3223-4_6

Altrabsheh, N., Cocea, M., & Fallahkhair, S. (2014). Learning sentiment from students' feedback for real-time interventions in classrooms. In *Adaptive and Intelligent Systems* (pp. 40-49). doi:10.1007/978-3-319-11298-5_5

Anjewierden, A., Kolloffel, B., & Hulshof, C. (2007, September). *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 (ADML 2007), Crete, Greece.

Baker, D. P., Horvarth, L., Campion, M., Offermann, L., & Salas, E. (2005). The ALL teamwork framework. In *International adult literacy survey, measuring adult literacy and life skills: New frameworks for assessment* (Vol. 13, pp. 229-272). Ontario, Canada: Statistics Canada.

Clark, E., & Araki, K. (2013). Turing test-based evaluation of an experimental system for generation of casual English sentences from regular English input. *JACIII, 17*(3), 353-361.

Crowston, K., Allen, E. E., & Heckman, R. (2012). Using natural language processing technology for qualitative data analysis. *International Journal of Social Research Methodology, 15*(6), 523-543.

Dennis, A. R., & Garfield, M. J. (2003). The Adoption and use of GSS in project teams: Toward more participative processes and outcomes. *MIS Quarterly, 27*(2), 289-323.

Dong, H., Cheung Hui, S., & He, Y. (2006). Structural analysis of chat messages for topic detection. *Online Information Review, 30*(5), 496-516.

Erkens, G., & Janssen, J. (2008). Automatic coding of dialogue acts in collaboration protocols. *International journal of computer-supported collaborative learning, 3*(4), 447-470.

Finkel, J. R., Grenager, T., & Manning, C. (2005). Incorporating non-local information into information extraction systems by Gibbs sampling. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics* (pp. 363-370). Stroudsburg, PA: Association for Computational Linguistics.

Han, B., Cook, P., & Baldwin, T. (2012). Automatically constructing a normalisation dictionary for microblogs. In *Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning* (pp. 421-432). Stroudsburg, PA: Association for Computational Linguistics.

He, W. (2013). Examining students' online interaction in a live video streaming environment using data mining and text mining. *Computers in Human Behavior, 29*, 90–102.

Hedayati, M., & Foomani, E. M. (2015). Learning style and task performance in synchronous computer-mediated communication: A Case study of Iranian EFL learners. *Educational Technology & Society, 18*(4), 344–356.

Koh, E., Hong, H., & Seah, J. (2014). An Analytic frame and multi-method approach to measure teamwork competency. In *Proceedings of the 14th IEEE International Conference on Advanced Learning Technologies* (pp. 264-266). doi:10.1109/ICALT.2014.82

Koh, E., Shibani, A., & Hong, H. (2016, June). *Teamwork in the balance: Exploratory findings of teamwork competency patterns in effective learning teams.* Paper presented at the 12th International Conference of the Learning Sciences (ICLS 2016), Singapore.

Kotsiantis, S. B., Zaharakis, I., & Pintelas, P. (2007). Supervised machine learning: A Review of classification techniques. In *Frontiers in Artificial Intelligence and Applications* (Vol. 160, pp. 3-24). Amsterdam, The Netherlands: IOS Press.

Powers, D. M. W. (2011). Evaluation: from precision, recall and f-measure to ROC, informedness, markedness & correlation. *Journal of Machine Learning Technologies, 2*(1), 37-63.

Romero, C., López, M.-I., Luna, J.-M., & Ventura, S. (2013). Predicting students' final performance from participation in online discussion forums. *Computers & Education, 68,* 458–472.

Rosé, C., Wang, Y.-C., Cui, Y., Arguello, J., Stegmann, K., Weinberger, A., & Fischer, F. (2008). 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.

Rychen, D. S. E., & Salganik, L. H. E. (2003). *Key competencies for a successful life and a well functioning society*. Cambridge, MA: Hogrefe & Huber Publishers.

Shibani, A., Koh, E., & Hong, H. (2015). Text mining approach to automate teamwork assessment in group chats. In *Proceedings of the Fifth International Conference on Learning Analytics And Knowledge* (pp. 434-435). New York, NY: ACM.

Siemens, G., & Gašević, D. (2012). Special issue on learning and knowledge analytics. *Educational Technology & Society, 15*(3), 1–2.

Villatoro-Tello, E., Juárez-González, A., Escalante, H. J., Montes-y-Gómez, M., & Pineda, L. V. (2012, September). *A Two-step approach for effective detection of misbehaving users in Chats*. Paper presented at the Conference and Labs of the Evaluation Forum (CLEF), Rome, Italy.

Appendix

Table A1. Teamwork competency coding scheme examples (Koh et al., 2014; Koh et al., 2016)

Teamwork dimension	Examples
Coordination (COD)	
Organize activities to complete task on time	"Faster!"
Ask team members who is in the team	"Who is in our group?"
Coordinate logistics of task	"Combine the answer"
Mutual Performance Monitoring (MPM)	
Give clarifying feedback to help in the team's performance	"Shall we go and search for ideas?"
Ask team members to contribute to the task	"What do you think?"
Steer conversation back to task	"Ok back to the discussion"
Team Decision Making (TDM)	
Give ideas related to the task	"Close down the factory"
Ask any task-related question	"What is most important?"
Exchange information about the task	"The factory can be moved to another location."
Constructive Conflict (CSC)	
Explain and give reason for disagreement	"I don't completely agree because …"
Add on to ideas	"Also, it is very expensive to move."
Propose different ideas	"How about adding filters to the factory"
Team Emotional Support (TES)	
Greet and introduce oneself	"Hi!"
Express positive emotions and emoticons	"yay! ☺"
Appreciate team member	"Good job Peter"
Team Commitment (TCM)	
Express confidence in own team's ability	"We have the longest chat"
Show togetherness through 'We' language	"Let us do it"
Hold own team in higher regard	"We are better"

Table A2. Rules defined to isolate nc lines

Spam type	Examples
Spamming with emoticons, characters and sentences	:dislike::dislike:", "ttttttttttttttttttt",
Spamming with random characters or other languages	"sdf jkl jk", "l Ã¯Â¼Â£ Â", "jadi sekarang melakuka"
Talking about other teams or their physical locations	"im in the other room", "John's group is talking about pokemon"
Discussion about random stuff out of task or classroom matters	"who likes gangnam style"
Talking to the admin or technical issues	"Chat Admin tell us who are u???", "its very slow and lagging"

Table A3. List of punctuations tagged

Punctuations	Tags	Reason
Positive emoticons (☺, :like:, :D)	{{pos_emo}}	Shows positive support to team member
Neutral and negative emoticons (☹, :o)	{{neut_emo}}	Do not contribute to teamwork dimensions
@	{{ref_mark}}	Used to tag or refer to a person in a chat in the context of mutual performance monitoring.
?	{{question_mark}}	Helps to identify if a chat line is a question or an agreement
*	{{asterisk_mark}}	Used by students at times to show typo correction which indicates improvement of the answer.
:	{{colon_mark}}	Used by students to consolidate and present the answer to the problem showing co-ordination.
Hyperlinks http://	{{weblink}}	Instances of providing external references related to the task like websites found in web search