Towards educational data mining: Using data mining methods for automated chat analysis to understand and support inquiry learning processes

Anjo Anjewierden, Bas Kollöffel, and Casper Hulshof

Department of Instructional Technology, Faculty of Behavourial Sciences,
University of Twente, PO Box 217, 7500 AE Enschede, The Netherlands
{a.a.anjewierden,b.j.kolloffel,c.d.hulshof}@utwente.nl

Abstract. In this paper we investigate the application of data mining methods to provide learners with real-time adaptive feedback on the nature and patterns of their on-line communication while learning collaboratively. We derived two models for classifying chat messages using data mining techniques and tested these on an actual data set [16]. The reliability of the classification of chat messages is established by comparing the models performance to that of humans. Results indicate that the classification of messages is reasonably reliable and can thus be done automatically and in real-time. This makes it, for example, possible to increase the awareness of learners by visualizing their interaction behaviour by means of avatars. It is concluded that the application of data mining methods to educational chats is both feasible and can, over time, result in the improvement of learning environments.

1 Introduction

Streifer and Schumann [22] describe data mining as: "a process of problem identification, data gathering and manipulation, statistical/prediction modelling, and output display leading to deployment or decision making" (p. 283). Luan [13] has argued that in (higher) education, data mining can have an added scientific value in fostering the creation and modification of theories of learning. This paper discusses first steps towards an integration of data mining and computer-supported collaborative learning (CSCL) to guide learners.

Theoretical and technological advances in the past decades have promoted new views on learning. Two modern concepts are the constructive nature of learning and its situated character [17]. The first concept argues that learners are in control of their own learning process and 'construct' personal knowledge. The second concept stresses that knowledge construction cannot occur in vacuo. The learning situation, that is the presence of tools and other learners mediate the knowledge construction process [24]. These concepts have spawned new instructional strategies, most importantly scientific inquiry learning and CSCL [18]. Computer-based simulations facilitate the implementation of appropriate learning environments to promote both types of learning [6]. Educational simulations model phenomena. They allow learners to explore and experiment with a

virtual environment, in order to discover the underlying properties of the simulation's behavior. A particular feature of computer-based simulations is that all user actions can be kept track of (or 'logged') [8]. Monitoring user actions can be used for feedback to learners about their rate of progress, or for adjusting instructions to individual learners [9]. Monitoring user actions can also be used to provide feedback in a CSCL context, for example to guide collaboration or communication. There are many types of CSCL environments. An interesting type is an environment where learners work simultaneously on the same task, but from physically separate locations. In such a case, communication usually proceeds through a text-based online chat interface.

Online chatting differs in a number of ways from everyday face-to-face conversation, both qualitatively and quantitatively. In chatting, learners tend to be more succinct, to focus more on technical and organizational issues instead of domain aspects, and to easily jump from topic to topic which makes for an erratic conversational pattern [23]. This can have positive effects (e.g., brainstorming), but also detrimental when the situation requires learners to focus on one topic [12]. In the latter case, there is a need for tools that help learners to focus, by aggregating, organizing, and evaluating the informational input by group members. An example is a tool developed by Janssen, Erkens, Kanselaar, and Jaspers [11], that could visualize the (quantitative) contribution of individual members to a group discussion in a CSCL environment. They found that use of the tool affected the communication style. For example, learners who used the tool wrote lengthier messages.

Our exploration tries to improve on the (visual) feedback on collaborative processes, and to take it a step further. The goal is to provide learners with feedback on the nature and patterns of their communication. For example, the above reported finding that during online chatting learners focus more on regulative than domain-related issues, can be monitored and utilized to give appropriate feedback on a just-in-time basis. Since learners cannot be expected to oversee the whole of their communication process, guidance will be invaluable. The practical issue that continuous presence of a tutor or teacher is very laborious and expensive may be solved by the integration of a guiding tool in the CSCL environment. Of course, in order to provide appropriate feedback and guidance to learners, the tool needs to be able to identify the nature and contents of the messages posted by the learners, and of communication patterns in general. To achieve this, a common step in the analysis of communication patterns is to define functions for different types of messages. We have found the distinction into four functions made by Gijlers and de Jong [7] suitable for our purposes. They distinguish between transformative (domain), regulative, technical, and off-task (social) messages. Domain (or transformative) messages concern expressions referring to the domain at hand, as long as these messages are not of a regulative nature. Regulative messages relate to planning or monitoring of the learning process. Technical messages concern the learning environment itself, tools, hardware, and software. The fourth category comprises social messages like greetings, compliments, remarks of a private nature, and so on.

Overview Our goal in developing an automated chat analysis tool is to apply data mining approaches to the problem of classifying chat messages. Section 2 gives an overview of the methods used.

Section 3 contains the result of applying the methods on an actual data set, based on data collected by Nadira Saab for her PhD research [16]. Saab's research focused on the role of support and motivation in a collaborative discovery learning environment, and on the communicative activities that can be found in such an environment. The experimental setup that was used involved pairs of secondary school learners, who worked together with a computer-simulated learning environment called *Collisions*. During learning, learners worked collaboratively with a shared interface, communicating through a chat message box. These messages, among other learning activities, were logged.

In order to determine the reliability of the method's assignment of functions to messages, its performance was compared to that of human raters. The goal of automated chat analysis is to build a new support tool to assist learners in CSCL environments. An example of such a use, which makes use of simple *avatars*, is given in Section 4.

2 Methods

A common step in the analysis of communication patterns is to define functions for different types of messages. A message is conceived here as "a series of words with a single communicative function" (cf. [7], [10], [11]). Most chat messages are very short and contain only one function. In other instances, messages can be segmented on the basis of for example, punctuation marks (e.g., full stop, question mark, exclamation mark, comma) and connectives (e.g., 'and', 'but') [10], [11]. The next step is to assign tags to each message, indicating its function. In the present study four functions are distinguished: regulatory, domain, social, and technical (cf. [7], also see Section 1). It is recommended to define functions rather broadly. More fine-grained definitions will increase the number of functions that are to be distinguished and will decrease the average frequency of observations within each category, which yields data that is (a) too detailed to be very informative and (b) hard or impossible to analyze statistically [5]. In the present study we are mainly interested in classifying each chat message as regulatory, domain, social or technical. In addition, we require the classification to be automatic in order to be able to give real-time feedback to learners.

A general method for classification is to define a set of features that can be extracted from an item (a chat message in our case) and then derive a model which, given the features of a particular item, can determine the (correct) classification.

Given that messages are natural language, the features have to be derived from syntactic patterns that occur in natural language. The simplest pattern is a single word (or possibly a compound term). For our chat corpus words like "speed", "momentum", "increases", "constant" point to domain oriented chats. More complex patterns can be defined by including generalisations. For example, "what ... think"[1], "the answer is ... #" (# is a number) point to regulatory messages. Some grammatical patterns also have a strong tendency to point to a certain class. For example, the vast majority of chats matching the pattern "`<uh> <uh>`" (`<uh>` is shorthand for an interjection, see Appendix A) are regulatory, whereas "`<at> <nn> is`" points to domain-oriented messages such as "the speed is increasing".

[1] We use the pattern syntax of tOKo [2].

We experimented with two automated methods for the classification of messages based on the following features:

Words A common approach is to consider a document as a bag-of-words and use word occurrence as a feature. Although historically, this approach has been mainly used to distinguish between topic-oriented classes (e.g., documents on cats and cars), it appears reasonable to assume regulatory chats contain different words than domain chats. The model results in the probability for a word (the feature) to belong to a given class.

Shallow grammars For chats, and particularly for the classes in our study, it is likely that the grammatical structure of a message is a reasonably strong indicator of the class. Regulatory messages, "ok, I agree", are different from domain oriented messages ("the speed increases") not only by the words they contain but also by their grammatical structure. Part-of-speech (POS) taggers can generalise natural language to a grammatical pattern in which each word is assigned the grammatical function it plays in the sentence: "ok/uh, I/pp, agree/vb", "the/at speed/nn is/vb constant/jj" (the symbol after the slash is the assigned POS-tag). The grammatical pattern can then be used as a feature for classification.

2.1 Data normalisation

Of the raw data collected by Saab [16] we used 78 chat sessions, containing 16879 chat messages in total. Most of the chats are in Dutch, or perhaps more accurately a derivative of Dutch emerging from the use of messaging tools, and a small fraction of English ("we are the greatest").

The corpus poses two significant challenges for automated analysis: it is very noisy and the messages are short. A total of 5749 different words were found in the raw data, of these 3353 (58.3%) are not given in the Dutch dictionary [3] we used. 8223 messages (48.7%) contained at least one unknown word. The distribution of the number of words over the messages was: 389 (0 words; an integer, punctuation only, smileys), 5502 (1 word; ok, yes, no, etc.), 3008 (2), 2857 (3), 2300 (4), 1669 (5), 852 (6), 259 (7), 36 (8), 7 (9).

The noisiness of the data is caused by several factors: misspellings, compounding, chat language, abbreviations and initialisms ("answ" for answer), reduplications ("heeellllooo'), and frivolous spellings of interjections ("okey"). Such noise can to some extent be corrected semi-automatically as it affects only single words [25].

A class of noise that is nearly impossible to correct automatically is when the specific context is relevant and, even worse, when multiple words in a message make it noisy. Consider the messages "k dan" (*okay, agreed*) and "k ook" (*me too*). In the chats the letter k is often used as an *abbreviation* for **ok** and for **ik** (I; first person pronoun). The correct spelling is therefore "oké dan" and "ik ook". Other examples are: 'ksnap t" ("ik snap het", *I understand*) and "kheb geni dee' ("ik heb geen idee", *I have no idea*).

We normalised the chats using a two stage process. First, we used noise correction methods in tOKo [2] to get rid of most misspellings, compounds and some reduplications. Next, we manually corrected nearly 3000 other errors in the chats. After normalisation, the chats contained 2323 different words of which 789 (33.9%) are not in the

dictionary. Most of the unknown words remaining have a (very) low frequency. The normalised data was used for the study.

2.2 Experimental setup

In order to train the algorithms for the automatic methods (bag-of-words and shallow grammars) four test sets of 400 messages were randomly selected from the corpus. The random selection process was biased towards longer messages to obtain a reasonable distribution over the four classes. Most messages are short and short messages tend to be regulatory.

Each set of test messages was scored by a researcher from our department, with the following options: one of the classes, other, and for ambiguous messages the option to score a message as belonging to multiple classes. After training, the coders took about 20 minutes to score their set of 400 messages.

The set of messages used in the experiment comprises those consistent among the raters. Given that most of the messages were rated by a single person, an expert was asked to check the classifications. In less than 1% of the cases, she might have used a different assignment.

2.3 Feature extraction

The features for the word method are the words themselves, integers, smileys and the question mark and exclamation mark. If a message contained a feature multiple times only one occurrence counted. For example, “the answer is 4! :-)” results in the feature set (answer, is, the, #, !, :-)) where # is any integer.

For the shallow grammars TreeTagger [20] was used to POS-tag the entire corpus. The resulting tag sequences were then input to the apriori algorithm [1] to determine the longest sequences that occurrred at least 20 times. This resulted in 546 POS sequences. Each coded message was also run through the POS-tagger and all non-overlapping POS sequences in the set of 546 it contained were taken as grammar features for that particular message.

2.4 Model construction and classification

Of the published methods for text classification, models that make the naive Bayes assumption of the features being independent have experimentally performed well compared to more sophisticated and computationally more expensive methods [15] (see [14] for an overview of alternate methods). Naive Bayes classifiers, in the context of text classification, are normally applied to entire documents which introduces issues of both feature selection and feature weights (frequencies). In the context of chat message classification such issues do not play a role.

Each message is represented as a feature vector $F = (f_1, f_2, ..., f_n)$ were f_k is 1 if the feature (word or grammar sequence) is present in a chat message and 0 when not. The conditional probability of feature f_k belonging to class C_i is then: $p(f_k|C_i) = \frac{s_k}{s_i}$ were s_k is the number of coded messages assigned to class C_i that have f_k as a feature and s_i the total number of coded messages assigned to class C_i.

A message can be classified by selecting the class that has the highest value for the product of the conditional probabilities of the features it contains:

$$argmax_i P(F|C_i) = \prod_{k=1}^{n} p(f_k|C_i)$$

Several others (e.g., [15]) have observed that there is a problem when applying the above function to text classification because not all data items contain all features. Consider the message "well done honey" (C_x = social). When the feature "honey" does not occur in domain-oriented messages then $p(honey|domain) = 0$. Given the creativity of chatting learners someone is likely to come up with a message reading "honey, the speed increases". Substituting $p(honey|domain) = 0$ in the above function results in $p(F|domain) = 0$, which clearly is undesirable.

The solution we opted for is to assign a *minimal* probability to a feature independent of class. In other words, we assume that any feature not observed in the training set, has an equal probability of appearing as a feature in any class. The minimal probability is the following constant:

$$p(f_k|C_i) = 1/\sum_i s_i$$

This probability is larger than 0 and lower than any observed probability in the training set.

3 Results

Table 1 shows the results of applying the feature models as a classifier compared to the coded messages. The rows contain the human-coded messages and the columns the classification of the model. The values on the long diagonal are agreement between coders and the model (e.g., 834 messages are assigned to the regulatory class by both the coder and the word model).

Words	R	D	S	T		**Grammar**	R	D	S	T	
Regulatory	834	44	3	5	886	Regulatory	802	46	31	7	886
Domain	23	167	0	1	191	Domain	39	147	4	1	191
Social	51	2	113	1	167	Social	75	9	81	2	167
Technical	1	1	0	34	36	Technical	8	2	2	24	36
	909	214	116	41	1280		924	204	118	34	1280

Table 1. Classification of messages for coders (rows) and feature (columns) models: words (left), grammar (right)

An example of interpreting the table is to look at the first row and the second column. 44 messages were coded as regulatory and classified as domain-oriented. Cohen's

kappa [4] can be used to quantify agreement between coders and the model. The formula is:

$$\kappa = \frac{A - D}{N - D}$$

Here A is agreement (sum of the values on the long diagonal), D is disagreement (the other values) and N the total number of items scored. For the word model $\kappa = 0.88$ which is considered as a good interrater reliability (> 0.8) in the social sciences. $\kappa = 0.79$ for the grammar model.

There are several things to consider. The most important is that the difference between the classes is not well-defined. In general, the domain class is the most easy to identify by humans as it, more or less by definition, requires the presence of some domain specific terms. A *correct* classification of the other classes often depends on reference to the previous messages which neither the human coders nor the classifier had access to. Generally, the distinction between the regulatory and the social classes is very subtle. Coders were instructed to classify a message as social when it contained a positive or negative social term ("ok, continue" is regulatory, whereas "ok, nerd" is social).

In Section 2 we hinted at the distinction between *defining* patterns in messages and *discovering* patterns. Inspecting the two classification models makes it possible to informally determine whether the approach we followed results in the automatic discovery of patterns (terms or shallow grammars).

Some examples of terms discovered as "belonging" to a particular class ($p > 0.8$) by the word model are: **domain**: mass, v (velocity), constant, collision, axis, increases; **regulatory**: question, understand, wait, next, correct, try, look, ok, idea, seems, etc.; **social**: stupid, fun, nerd, nice, hi; **technical**: mouse, window, program, pointer, logged.

Similarly, and perhaps surprisingly, the grammar model also discovers syntactic structures that are significant for a single class. Examples for the domain class are: "<nn>/<nn>" (m/s), "<at> <nn> <vb> <jj>" (the speed is larger) and for the regulatory class "<vb> <pp>" or "<vb> <uh>" for example are significant.

4 Application

In the previous sections we have described an approach to automatically classify educational chat messages as regulatory, domain-oriented, social and technical. The approach can be used to assist researchers with their analysis. A slightly more ambitious application is to provide learners with real-time adaptive feedback on their behaviour. One idea is to display an avatar of the learner which dynamically depicts the *ratios* of messages classified. Such avatars could increase learner awareness, for example by providing "subtle" hints to learners that they should focus more on the domain. provides an example.

The correspondence between the avatars and the classification is as follows: body (regulatory), head (domain), arms (social) and legs (technical). The learner avatar on the left in Fig. 1 has a large body and a small head, indicating s/he chats too little about the domain. In contrast, the avatar on the right chats more about the domain and uses

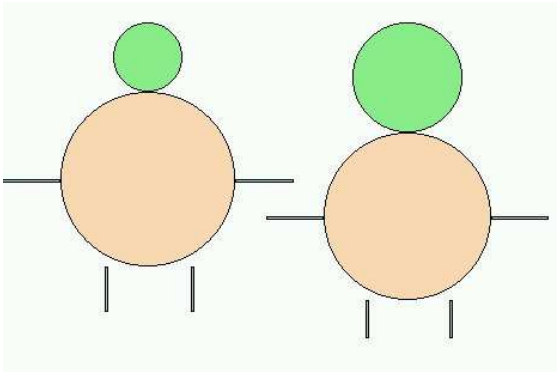

Fig. 1. Learner avatars derived from the classification

fewer regulatory messages. Note that we are mainly interested in the ratio of domain and regulative messages. Both technical and social messages can be considered off-task.

Others have used simple measures, for example the number of chat messages as an indicator of participation. The avatars provide a more sophisticated form of feedback: an indication of what the learners are discussing in the learning environment.

5 Discussion and conclusions

Results suggest that the classification of messages is reasonably reliable and can be done automatically and in real-time. We believe that this provides an interesting opportunity to improve learning environments.

Several practical issues remain. The most important one is the ability of the classifier to "understand" a message as it is typed. As mentioned in Section 2.1 the data we used was extremely noisy and automatic noise correction appears beyond the state of the art. The implication is that learners have to be teased to type more carefully. Another issue is that the method requires key (domain) terms of the learning environment are understood by the avatar. For most inquiry learning environments these terms are known in advance and they can be given an estimated conditional probability if not enough training data for the model is available. We do not expect a large difference in the vocabulary or grammar for regulatory messages. A cursory analysis of chat data from another learning environment confirms this.

An alternative to the automatic classification of messages is the manual definition of terms and syntactic patterns. We have investigated this by developing an ontology of terms related to each of the four classes and syntactic patterns (mainly for the regulatory class). The outcome of the word and grammar models can be used to further refine these "semantic" classifications. A formal evaluation of the manual approach is hardly possible as many chat messages don't match any of the terms or patterns. The avatars, however, exhibit similar shapiness for both the manual and automatic approaches.

In this paper we have considered chat messages in isolation. To understand the meaning of the communication this is clearly not sufficient. In several cases, even for

our four classes, a message can only be classified correctly when the previous messages are taken into account. For example, "4, I think" could be domain oriented when "4" refers to a value of a variable and regulatory when it refers to an answer. The analysis of sequences of chat messages, for example [21] who used Hidden Markov Models to analyse already coded chats, in combination with semantic analysis is therefore a possible direction for a more detailed understanding of chat content.

We conclude that the application of data mining methods to educational chat data is feasible. For this paper we have restricted ourselves to the analysis of the chats only, in the future we plan to also look at the relation between what learners are saying and what they are doing in the learning environment.

Acknowledgements

We would like to thank Nadira Saab for making the chat data available, Hannie Gijlers for advice, encouragement, and help with the experiments, and Petra Hendrikse, Sylvia van Borkulo, Jan van der Meij and Wouter van Joolingen for switching from their usual role as a researcher to that of a subject, and the anonymous reviewers for there extensive and constructive comments. This research was funded by a grant from the Institute of Behavioural Research at the University of Twente.

A Part-of-speech tags

The part-of-speech tags used by the grammar model are based on the guidelines of the Penn Treebank project for English [19].

at	article	pn	pronoun
cc	coordinating conjunction	pp	personal pronoun
cd	cardinal number	ppd	possessive pronoun
dt	determiner	rb	adverb
in	preposition	uh	interjection
jj	adjective	vb	verb
nn	noun	wp	interrogative pronoun
od	ordinal number		

References

1. A. Agrawal and R. Srikant. Fast algorithms for mining association rules. In *International Conference on Very Large Databases*, pages 487–499, Santiago, Chile, September 1994.
2. A. Anjewierden et al. tOKo and Sigmund: text analysis support for ontology development and social research. http://www.toko-sigmund.org, 2007.
3. R. H. Baayen, R. Piepenbrock, and L. Gulikers. The CELEX lexical database (release 2) [cd-rom]. Philadelphia, PA: Linguistic Data Consortium, University of Pennsylvania, 1995.
4. J. Cohen. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20:37–46, 1960.
5. F. de Jong, B. Kollöffel, H. van der Meijden, J. Staarman, and J. Janssen. Regulative processes in individual 3d and computer supported cooperative learning contexts. *Computers in Human Behaviour*, 21:645–670, 2005.

6. T. de Jong. Technological advances in inquiry learning. *Science*, 321:532–533, 2006.
7. H. Gijlers and T. de Jong. The relation between prior knowledge and students' collaborative discovery learning processes. *Journal of Research in Science Teaching*, 42:264–282, 2005.
8. C. D. Hulshof. Log file analysis. In *Encyclopedia of Social Measurement*, volume 2, pages 577–583, Manchester, UK, 2004. Elsevier.
9. C. D. Hulshof and T. de Jong. Using just-in-time information to support scientific discovery learning in a computer-based simulation. *Interactive Learning Environments*, 14:79–94, 2006.
10. J. Janssen, G. Erkens, and G. Kanselaar. Visualization of agreement and discussion processes during computer-supported collaborative learning. *Computers in Human Behaviour*, 23:1105–1125, 2007.
11. J. Janssen, G. Erkens, G. Kanselaar, and J. Jaspers. Visualization of participation: Does it contribute to successful computer-supported collaborative learning?
12. D. S. Kerr and U. S. Murthy. Divergent and convergent idea generation in teams: A comparison of computer-mediated and face-to-face communication. *Group Decision and Negotiation*, 13:381–399, 2004.
13. J. Luan. Data mining and knowledge management in higher education: Potential applications. In *Annual Forum for the Association for Institutional Research*, Toronto, Ontario, Canada, June 2002.
14. C. D. Manning and H. Schütze. *Foundations of Statistical Natural Language Processing*. The MIT Press, Cambridge, Massachusetts, 1999.
15. A. McCallum and K. Nigam. A comparison of event models for naive bayes text classification. In *AAAI-98 Workshop on Learning for Text Categorization*, Madison, Wisconsin, July 1998.
16. N. Saab. *Chat and Explore: The role of support and motivation in collaborative scientific discovery learning*. PhD thesis, University of Amsterdam, 2005.
17. G. Salomon and D. N. Perkins. Individual and social aspects of learning. *Review of Research in Education*, 23:1–24, 1998.
18. H. Salovaara. An exploration of students' strategy use in inquiry-based computer-supported collaborative learning. *Journal of Computer Assisted Learning*, 21:39–52, 2005.
19. B. Santorini. Part-of-speech tagging guidelines for the Penn Treebank project. http://www.cis.upenn.edu/˜treebank, 1991.
20. H. Schmid. Probabilistic part-of-speech tagging using decision trees. In *International Conference on New Methods in Language Processing*, Manchester, UK, 1994.
21. A. L. Soller. *Computational analysis of knowledge sharing in collaborative distance learning*. PhD thesis, University of Pittsburgh, 2002.
22. P. A. Streifer and J. A. Schumann. Using data mining to identify actionable information: Breaking new ground in data-driven decision making. *Journal of Education for Students Placed at Risk*, 10:281–293, 2005.
23. H. I. Strømsø, P. Grøttum, and K. H. Lycke. Content and processes in problem-based learning: A comparison of computer-mediated and face-to-face communication. *Journal of Computer Assisted Learning*, 23:271–282, 2007.
24. J. van der Linden, G. Erkens, H. Schmidt, and P. Renshaw. Collaborative learning. In R. J. Simons, J. van der Linden, and T. Duffy, editors, *New Learning*, pages 37–54, Dordrecht, 2000. Kluwer Academic.
25. W. Wong, W. Liu, and M. Bennamoun. Integrated scoring for spelling error correction, abbreviation expansion and case restoration in dirty text. In *Australian Conference on Data Mining*, Sydney, Australia, 2006.