

# Typicality in Computer Mediated Discussions

## — An Analysis with Neural Networks —

Michael R. Berthold and Fay Sudweeks

**Abstract**— ProjectH, a large group of international researchers, produced a huge amount of data from computer mediated discussions. The data classified several thousand postings from more than thirty newsgroups. One approach to extract typical messages from this database is presented in this paper. An autoassociative neural network was trained on 3000 coded messages and then used to construct typical messages under certain specified conditions for several scenarios. This paper illustrates the architecture of the neural network that was used and explains the necessary modifications to the coding format. In addition several “typicality sets” produced by the neural net are shown and their generation is explained. In conclusion the ANN is used to explore the types of messages that typically initiate or contribute to longer lasting threads.

### I. INTRODUCTION

As computer networks expand into homes and organisations, and high-speed network highways provide a medium for communication and community formation on a scale that has never been feasible before, new mores are created. People invest varying amounts of time and energy in communicating on computer mediated discussion groups with people they have mostly never met face-to-face. The communication within these groups varies along a continuum of interrelatedness.

Michael R. Berthold, University of Karlsruhe, Institut für Rechnerentwurf und Fehlertoleranz (Prof. D. Schmid), Postfach 6980, 76128 Karlsruhe, Germany. eMail: berthold@ira.uka.de  
Fay Sudweeks, Key Centre of Design Computing, University of Sydney, Sydney, NSW 2006, Australia. eMail: fays@arch.su.edu.au

Rafaeli [4, 5], Rogers and Rafaeli [8] and Rafaeli and Sudweeks [6] argue that the variable that affects the interactive nature of computer-mediated communication (CMC) is the theoretical construct of interactivity — the extent to which messages in a sequence relate to each other, and especially the extent to which later messages recount the relatedness of earlier messages.

In this paper we use an autoassociative neural network (ANN) to analyse and explore communication density, the extent to which messages in a thread cohere. The data set comprises 3000 postings to 30 newsgroups classified on 46 variables or groups of features. In the context of categorisation, each variable equates with a reference point or feature within some information setting. We propose that the ANN is capable of identifying the features of messages that typically initiate or contribute to longer lasting threads. Our findings support the construct of interactivity as a variable of communication settings. We also demonstrate that an ANN is a valuable preprocessor to other analytical methods.

### II. THE DATA

#### A. Preprocessing

The data set was created by ProjectH [10], a large group of researchers who collaboratively collected a representative sample of computer mediated discussions. More than a hundred people from fifteen countries used computer networks to plan, organise and implement a quantitative study of social and linguistic dynamics in public newsgroups and mailing lists. Batches of 100 messages were downloaded from publicly-accessible archives of randomly selected discussion groups on Internet, Bitnet and Compuserve

and coded on 46 variables (See [7] for a detailed description of the methodology).

In all, 4322 messages were coded of which 1000 were double coded for reliability purposes, 2000 were single coded, and 322 were partially coded batches. The partially coded batches were excluded, and one of each double coded list was chosen randomly resulting in a database of 3000 messages.

### B. Preparation for the ANN

The database was prepared for processing by an autoassociative neural network. First, identification variables (author-id, coder-id and message-id) were deleted. Second, the date and time variables were converted to two new entries, one indicating day of week and the other time of day (worktime, evening, night).

Third, three new entries were computed since the exploration of the nature of threads was a main focus of analysis:

- *reference-depth*: how many references were found in a sequence before this message.
- *reference-width*: how many references were found, which referred to this message.
- *reference-height*: how many references were found in a sequence after this message.

These entries were extracted from the original database, but were not present in individual entries, because they refer to sequences of references. Figure 1 shows a reference-tree to illustrate the used terms. The thread is the longest path from the top down into one of the branches in this tree, in the example this would be the path starting at A leading over B, E, G and K to L. Message E in this figure is being referenced directly by four messages (F, G, H and I) and results therefore in the *reference-width*=4, the same message E references a sequence of two messages (B and A), measured by the *reference-depth*=2 and is referenced by another sequence of three messages (G, K and L) leading to a *reference-height* of 3. Note that several of these messages could have been written by the same author. Different labels in this example only indicate different messages, not different authors.

The final list, now containing 51 entries per message, was recoded in a binary format for a

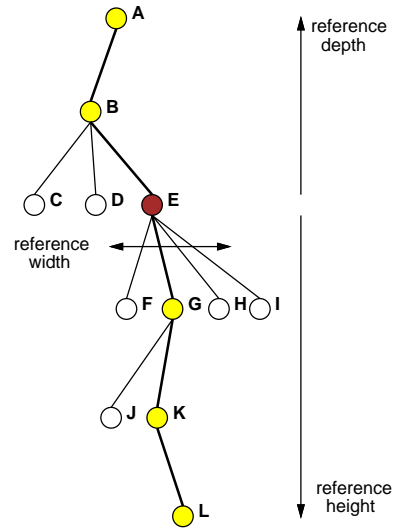


Figure 1: A reference-tree, illustrating the terminology. A thread starts with message A and the last message participating is L. Message E is being referenced directly by four messages (the *reference-width*), references itself a sequence of two message (*reference-depth*) and is referenced by a sequence of three messages (*reference-height*)

neural network (see table 1 for a few examples). Since the original coding scheme had several options per entry (such as 4 different options for the number of lines of a message (< 10, 10–25, 25–100 and > 100), each entry was split into as many *features* as the entry had options, i.e. a binary coding scheme. The recoding resulted in 149 binary features in the new database. A complete list of features can be found in [1].

### III. APPLYING AN ANN TO 3000 MESSAGES

Since the data consisted of 149 binary features, the resulting autoassociative neural network had 149 fully connected units with the usual sigmoidal activation function. No additional hidden units were used for the experiments reported here. This leads to  $149 * 149 = 22,201$  weights and 149 thresholds to adjust during training. Training was done using the well known Hebbian training procedure, with an additional stochastic annealing (see [9] for details). Almost 100 hours of CPU-time were spent before the error rate of the network started to settle on a plateau.

### A. Interpreting the weight matrix

A preliminary look at the weight matrix of the final network gives first interesting insights. Here only a few examples are listed:

- excitatory connections (positive weights):
  - the large positive weight connecting unit 1, corresponding with the feature LINES-A (1–10 lines of original text) and unit 13, OPINION-A (no opinion is stated), indicates a strong excitatory influence. This leads to the conclusion that short messages (1) state no opinion (13). However this does not mean that short messages always state no opinion, but it is a property of the database the network picked up. It is still possible that other units inhibit unit 13 much stronger and therefore it is not always going to be active if unit 1 is active.
  - weight  $1 \rightarrow 43$ : short messages (1) request information (43).
  - weight  $33 \rightarrow 1$ : unformatted messages (33) are short (1).
- inhibitory connection (negative weights):
  - weights  $54/55 \rightarrow 1$ : messages containing artistic icon(s) (54, 55) are **not** short (1), i.e. unit 1 receives strong negative feedback from activated units 54 and/or 55.

Of course, those observations can only reveal relations between a few of the networks’ units. To explore the dependencies between all of the 149 units, especially under certain conditions (usually modeled through clamped units), the network has to settle in a state with low energy. Clamping of units can be used to restrict the space the neural network is exploring to find a solution.

In the case of the ProjectH-ANN, the clamping technique was used to define certain properties of a message and to allow the network to determine which other features are correlated. This leads to the creation of *typical examples*

for specific features being **on**, discussed in the next section.

### B. Creating typical examples

To create a typical example, the feature(s) that are required to be present in the *feature set* are clamped and the network settles at the typical pattern. This worked well in the controlled examples described in [2] but, in the case of the ProjectH-ANN, there are several states the network can settle in. This is mainly due to the fact that the input data is not free of noise and not all of the units are going to be strongly correlated to other units (or correlated at all). Therefore the random update of the states will not always result in the same final state for the network. Instead one would expect the unaffected units to behave randomly and therefore the number of different states can be quite high.

But if the network is allowed to settle several times, each time starting from a different random starting state and using another random order for the asynchronous update, some features occur more frequently in the final states than others. Table 1 shows some activated features when feature 29 (message contains humour) was clamped.

This list only contains a few of the 149 total features, but it illustrates quite well how some features are strongly correlated to the feature that was clamped and others are not correlated at all. For example, non-humorous messages seem to be gender specific, since feature 58 (GENDER-C, male) is **on** 80% of the 20 experiments that were conducted. On the other hand, considering that almost 75% of all messages were written by males, the significance of this information is probably not very high. Also, non-humorous messages do not contain abusive language, as the strong response on feature 112 shows. Interesting is feature group 33–36, here none of the features has an exceptionally high occurrence. This leads to the assumption that a message containing humour does not depend on the formatting. This feature group seems to be not *significant* to feature 29.

This process can be done for all features separately or for combination of features clamped together. The result will be a list of features,

Table 1: The frequency of feature activations for feature no 29 (message contains humour).

no.	description	frequency of on activations
1	1-10 lines of real message	40%
2	11-25 lines of real mes.	50%
3	26-100 lines of real mes.	10%
4	> lines 100 of real mes.	0%
5	no subject line	5%
6	subject line is appropriate	95%
7	subject line is inappropriate	0%
...		
21	no question/request contained	70%
22	mes. contains question/request	30%
...		
28	no humour contained	0%
29	message contains humour	100%
...		
33	unformatted	25%
34	minimal formatted	35%
35	mostly formatted	35%
36	overformatted	5%
...		
56	can't tell gender of author	0%
57	female	20%
58	male	80%
...		
112	no abusive language	100%
113	abusive lang. about content only	0%
114	abusive lang. about person	0%
115	abusive lang. about general. others	0%
116	mixture	0%
...		

each with an indication of how often the network settled in a state which had this particular unit (or feature) being **on**. This information can be used to produced *typicality sets* as shown in the next section.

### C. Typicality sets of features

Since the main focus of analysis is correlations between features, it is interesting to extract a set of typical features from the output of the ANN. An a-priori specified threshold  $\Theta$  can be used to choose features for this set.  $\Theta$  determines how often a feature  $x$  has to occur in  $y$ 's feature set to be included in its typicality set (i.e.  $x$  is *typical* for feature  $y$ ).  $\Theta$  can also be interpreted as a minimum required correlation between  $x$  and  $y$ .

The example from the previous section is again used to show the resulting typicality set (see Table 2). This table shows which features are highly correlated with feature 29.

But so far there is no information about the differential quality of the list. It is certainly

Table 2: The typicality set for feature 29 (message contains humour) using  $\Theta = 80\%$ .

no.	description
6	subject line is appropriate
9	regular msg
18	no apology
26	no challenge/bet/dare
38	regular capitalization
53	no artistic icons
58	male
68	no quoted text from this list
72	no CMC text quoted from outside list
95	no first person plural
98	addresses other person
112	no abusive language
117	no intention to prevent/calm tension
120	no identification of status
126	no ending quotation
145	6pm-12am

possible that one or even several of these features appear in almost every typicality-set and are not well suited to distinguish between different message types. On the other hand, a feature could just behave randomly most of the time. It was therefore necessary to score the *sensitivity* of each feature.

### D. Scoring Features and Sensitivity

Of course, some of the features in the typicality set might not be as interesting as others because they are typical for almost all messages and therefore will be “**on**” no matter which feature is clamped. A feature behaving like this is called *insensitive*. To distinguish between sensitive and insensitive features, the features have to be ranked or scored in a way that indicates the sensitivity of the feature to the clamping of other features. This information is hidden in the distribution of features over all typicality sets and results in a *sensitivity score* for each feature. A low sensitivity indicates that the corresponding feature either appears or does not appear in almost all typicality set, a high sensitivity indicates an appearance that depends heavily on the clamped feature. Insensitive features can be seen as global characteristics of all observed messages.

## V. TYPICALITY IN CMC THREADS

The previous section described how typicality sets for single features can be generated by an autoassociative neural network. For an invest-

Table 3: Typical distinguishing features of a referenced (or “interactive”) message.

no.	feature description	score
2	11-25 lines of original text	1
6	subject line is appropriate	1
17	contains statement of a fact	1
21	no question/request	1
47	no emoticons	2
50	no punct device to express emotion	3
58	male	2
60	identifies gender via name/signature	3
72	no CMC text quoted from outside list	2
98	addresses other person	1

igation of CMC threads and their characteristics, some typicality sets are of more interest than others.

To analyse the nature of threads, we need to compare messages that start or continue a thread with messages that are not part of a thread. The reference-width variable was used to characterize an “interactive” versus a “non-interactive” message in the sense of participation in a thread. A message is called “interactive” if it is referenced by at least one other message; that is, it *participates* in a thread. In contrast a “noninteractive” message is not referenced at all; that is, it does *not participate* in a thread. Clamping the corresponding features (134 — no messages are referencing this message, and 135 — 1-2 references to this message) leads to two typicality sets for the two types of messages being investigated.

Interestingly the two typicality sets have several features in common. This is due to the fact that not every feature is sensitive to every other one. In addition some of the features have a low sensitivity score, meaning that they are not sensitive at all to other features. In creating the typical “interactive” and “noninteractive” message, features appearing in both typicality sets were deleted and features with a too low sensitivity score were discarded. This led to Tables 3 and 4 and finally enabled us to extract some properties of the messages in the database:

- An “interactive” message has medium length (2) and an appropriate subject line (6).
- A statement of a fact (17) enhances the chances of being followed-up.

Table 4: Typical features of a nonreferenced (or “noninteractive”) message.

no.	feature description	score
80	no prev msg referenced by this msg	4
87	new topic, no reference to prev discussion	18
95	no first person plural	1
131	no references after this msg	9

- If, during an already ongoing thread, one introduces a completely new topic (87), the chances of getting a response are slim. This point appears to be a very strong one because of the high sensitivity score of feature 87.
- Interesting also is that a message which does not refer to another message seems likely not to be referred to. But the sensitivity score of this feature is reasonably low, which makes sense, otherwise threads would never start. This discovery indicates, however, that the start of a thread with a new topic is not an easy task. Being followed-up when one already participates in a thread is much easier.

## VI. CONCLUSIONS

We have described a novel approach of using autoassociative neural networks to explore typicality in computer mediated discussions. We showed how to train an ANN and how the final weight matrix could be used to extract relationships between variables. We then used the neural network to define typicality sets for specified features and showed how messages which participate in threads (“interactive” messages) can be distinguished from those messages not participating in a thread (“noninteractive” messages).

This approach can be used to act as a preprocessor for a more detailed statistical analysis, concentrating on the subsets of features already discovered by the neural network. The ANN is used to discover feature groups that are correlated; further analyses would concentrate on the strength and statistical significance of those correlations.

In addition the approach presented here provides insights into the quality of the database. There are several blocks of features that are

Table 5: Sensitivity scores for some features (see [1] for full list).

feature	description	score
STYLE2-A	no colloquial spelling	5
STYLE2-B	contains colloquial spelling	2
DEPEND3-A	no ref to manner of prev ref	4
DEPEND3-B	ref to manner of prev ref	2
GENDER3-A	no gend spec terms re others	-55
GENDER3-B	gender spec terms regarding others	1

strongly correlated while other features are only loosely or not at all connected. In contrast to the example used by [3], noise from coder errors as well as differences on opinionated variables (as described by [6]) result in a database which is not as well structured as artificial ones.

The possibilities for applying an ANN to the data are far from exhausted; several other features are well worth exploring (a few examples are listed in table 5). STYLE2, for example, has high sensitivity scores for all features of the group. This would be another way of exploring the nature of threads. Also, an analysis about the quality of interactivity could be performed, by the DEPEND3 features, which describe the manner in which previous messages are referenced. Yet another example is GENDER3 which is also a feature group with high sensitivity scores for all features of the group. GENDER3 codes the fact that gender identification is an issue.

The ANN-approach we presented in this paper is obviously capable of extracting a form of relationships between features. It also added support to tentative hypotheses about computer-mediated communication and identified new directions for research.

## VII. ACKNOWLEDGMENTS

This work was supported by a University of Sydney Research Grant (URC). The authors would like to acknowledge many fruitful discussions with Richard Coyne and Sid Newton.

## REFERENCES

[1] Berthold, M. R., Sudweeks, F., Newton, S. and Coyne, R.: 1995, It makes sense: Using an Autoassociative Neural Network to explore typicality in computer mediated communication, in Rafaei, S., Sudweeks, F. and McLaughlin (eds), *Network and Netplay:*

*Virtual Groups on the Internet*, AAAI/MIT Press (to appear).

[2] Coyne, R. D. and Yokozawa, M.: 1992, Computer assistance in designing from precedent, *Environment and Planning B: Planning and Design*, **19**, 143-171.

[3] Coyne, R. D., Newton, S. and Sudweeks, F.: 1993, Modeling the emergence of schemas in design reasoning, in J. S. Gero and M. L. Maher (eds), *Modeling Creativity and Knowledge-Based Creative Design*, Lawrence Erlbaum, Hillsdale, New Jersey, pp. 177-209.

[4] Rafaei, S.: 1986, The electronic bulletin board: A computer driven mass medium, *Computers and the Social Sciences*, **2**(3): 123-136.

[5] Rafaei, S.: 1988, Interactivity: From new media to communication, in R. P. Hawkins, J. M. Wiemann and S. Pingree (eds), *Sage Annual Review of Communication Research: Advancing Communication Science*, Vol. 16, Sage, Beverly Hills, CA, pp. 110-134.

[6] Rafaei, S. and Sudweeks, F.: 1995, Interactivity on the Net, in Rafaei, S., Sudweeks, F. and McLaughlin (eds), *Network and Netplay: Virtual Groups on the Internet*, AAAI/MIT Press (to appear).

[7] Rafaei, S., Sudweeks, F., Konstan, J. and Mabry, E.: 1994, ProjectH overview: A quantitative study of computer mediated communication, *Technical Report*, University of Minnesota, MN.

[8] Rogers, E. M. and Rafaei, S.: 1985, Computers and communication, in B. D. Ruben (ed.), *Information and Behavior*, Vol. 1, New Brunswick, NJ, pp. 135-155.

[9] Rumelhart, D. E. and McClelland, J. L. (eds): 1987, *Parallel Distributed Processing: Exploration in the Microstructure of Cognition, Volume 1, Foundations*, MIT Press, Cambridge, Massachusetts.

[10] Sudweeks, F. and Rafaei, S.: 1995, How do you get a hundred strangers to agree: Computer mediated communication and collaboration, in T. M. Harrison and T. D. Stephen (eds), *Computer Networking and Scholarship in the 21st Century University*, SUNY Press, NY (to appear).