"It makes sense": Using an Autoassociative Neural Network to explore typicality in computer mediated discussions

Michael R. Berthold[1], Fay Sudweeks[2], Sid Newton[3] and Richard Coyne[2]

[1] Forschungszentrum Informatik
an der Universitaet Karlsruhe, Gruppe ACID (Prof. D. Schmid)
Haid–und–Neu–Strasse 10–14, 76131 Karlsruhe, Germany
[2] Key Centre of Design Computing
University of Sydney, Sydney, NSW 2006, Australia
[3] University of Western Sydney, Nepean
Kingswood NSW 2747, Australia

This paper is to appear in F. Sudweeks, M. McLaughlin and S. Rafaeli (eds) (1996), *Network and Netplay: Virtual Groups on the Internet*, AAAI/MIT Press. Please do not cite without permission.

Abstract. ProjectH, a research group of a hundred researchers, produced a huge amount of data from computer mediated discussions. The data classified several thousand postings from over 30 newsgroups into 46 categories. One approach to extract typical examples from this database is presented in this paper. An Autoassociative Neural Network is trained on all 3000 coded messages and then used to construct typical messages under certain specified conditions. With this method the neural network can be used to create "typical" messages for several scenarios. This paper illustrates the architecture of the Neural Network that was used and explains the necessary modifications to the coding scheme. 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 threads and the types of messages that typically initiate or contribute longer lasting threads.

1 Introduction

As the global mesh of computer networks expands at an exponential rate, reaching 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. The virtual groups that were largely a phenomenon of education and scientific institutions are populating other domains, and are changing communication practices and social structures.

The network is populated with people who invest varying amounts of time and energy in communicating on computer mediated discussion groups with other people they have mostly never met face-to-face. The groups that form thus vary along a continuum of communication interrelatedness. Why does this social phenomenon occur? More importantly, why do groups differ in communication styles?

Rafaeli (1986; 1988), Rogers and Rafaeli (1985) and Rafaeli and Sudweeks (1996) argue that the variable that affects the interactive nature of messages, threads and groups is the theoretical construct of interactivity—the degree to which communication transcends reaction. Interactivity is a pivotal measure of the social dynamics of group communication.

In this paper we use a connectionist model to analyse and explore the features of messages that typically initiate or contribute to longer lasting threads. 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. Our findings give further support to the construct of interactivity as a variable of communication settings.

1.1 Typicality

A key component of human thought is our ability to identify distinct categories or classes of information, which imposes order on an otherwise amorphous, continuous mass of sensory input.

The 'classical' view of categories is that they carve the world according to well-defined 'natural' boundaries (Smith and Medin, 1981; Pulman, 1983). Categories are defined in terms of necessary and sufficient features: all members of a category share necessary features, therefore entities that have sufficient necessary features of a category are members of that category. This classical theory of categorisation was established during the time of Aristotle and it is only recently that it has been questioned. In particular, experimental evidence during the 1970s demonstrated that members of a category vary in typicality—in how good an example is of its category. Categories boundaries then become ill-defined, if it all existing–what does exist are points of reference to which comparisons are made and which are combined in different ways depending on the particular context (Newton, 1992; Smith and Medin, 1981; Rosenman and Sudweeks, 1995).

Observed human behaviour displays a clear grading of membership which does not equate with the classical theory of categories. Rosch (1978), for example, claims that not all of the defining features of a category are necessary and that the more typical the example, the 'better' the membership of the category. The most typical members are referred to as prototypes. One way of representing a prototype is through an actual instance of a member, referred to as an exemplar. There is considerable evidence to support the notion of an exemplar (Kahneman and Tversky, 1973; Collins and Loftus, 1975) but it remains extremely difficult to formulate a structure for exemplars. How, for example, is the degree of correspondence (i.e. the similarity) between candidate and exemplar determined?

The most structural interpretation of similarity is based on features. The features are used to codify the known members, and act as a reference against which a decision about the membership of some candidate entity can be gauged. In its most trivial form, membership is determined on the basis of a candidate having a minimum 'threshold' number of features in common with the category representation. Interestingly, the list of features which form the summary representation of a category (the prototype) are not necessarily realisable as an actual instance. There may be no member of the category which matches the prototype on every feature. Thus we have a notion of non-necessary, modal features.

The major limitation of the approach is that membership is determined on the basis of some critical sum of the weighted features. The best-known applications of this process are the 'contrast' (Tversky, 1977) and 'spreading activation' or 'connectionist' (Collins and Loftus, 1975; Rumelhart and McClelland, 1987) models. In both cases, membership is determined on the basis of both similarity and dissimilarity. In the contrast model, ratings are summed statically; in spreading activation, weights are applied dynamically to coerce other features into play. Groups of features are formed, and these groups use their combined weights to force incompatible features out of consideration. In this sense, categories 'emerge' over several iterations, and final groupings represent the summary features of an implied category (Coyne et al., 1993). Categories are implied by the example entities used as input to the 'training' part of the connectionist approach.

Statistical analysis provides a suite of techniques that identify correlations existing between particular features in the data set. Statistical correlations then equate with the similarities between points that define a particular category within the information. The use of an incremental sum of squares analysis to pair highly correlated features provides an indication (typically represented in the form of a dendogram) of how individual features come together to form groupings of features. But these groupings are strictly hierarchical, and a more relevant interpretation is provided by a Euclidean form of cluster analysis (Everitt, 1974). A Euclidean cluster analysis attempts to separate out groups of features rather than, as is the case in hierarchical analysis, show their successive containment.

This form of multivariate statistical analysis provides a useful indication of where the aggregation (boundaries) within a given data set might appear. The characteristic features of a typical message can then be identified from the most cohesive clusterings of features. This form of analysis is widely recognised as providing a static view of the data—a 'snapshot' of the typical and atypical messages pertaining to a specific data sample. The clusterings are based entirely on pair-wise correlations. In human cognition the clusterings are more dynamically created across all features synchronously. As features are drawn into particular groupings they form dynamic allegiances which can effectively overrule the original cohesion based on a simple pair-wise correlation. This dynamic clustering is the effect to be explored in this paper.

The connectionist (or *associative neural network*) approach (Rumelhart and McClelland, 1987; Hertz et al., 1991; Mehra and Wah, 1992) exploits a distributed description of each particular message (instance) as a pattern of activation across all features (nodes). A particular clustering of features (category) emerges as the network stabilises on a particular pattern of activation. Each message is described in terms of features, such as relevance, time, tone and so on. The pattern of activation capture complex information about dependencies between combinations of features.

In identifying typicality in mediated discussions, a profile emerges of the features of messages that engage the attention of others, encourage participation, and predict the formation and/or maintenance of interactive communication settings.

2 The Data

2.1 Preprocessing

The data set was created by ProjectH (Sudweeks and Rafaeli, 1995), 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 randomly selected discussion groups on Internet, Bitnet and Compuserve and coded on 46 variables.[4]

Unix scripts were used to precode each batch of 100 messages on mechanical variables (e.g. 'coder-id', 'list-id', and 'author-id') and sent to coders to complete the remaining 40 variables. To accommodate a range of skills and technology, code forms were developed for different platforms. On completion, coders returned the full batch of coded messages for automatic processing. Each code form had an awk script to check syntax and to convert the data into a format suitable for most statistical packages. Other awk scripts checked field values for validity and consistency, and returned records with errors (e.g. missing values, values out of coding range, wrong message numbers, non-numeric codes). Error-free records were saved in a directory and the sender notified the number of records completed and outstanding. On completion of 100 error-free messages, the records were added to the database.

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.

[4] See (Rafaeli et al., 1994) for a detailed description of the methodology

2.2 Postprocessing

The database was converted to a form ready for processing by a Neural Network. First, the author–id, coder–id and message–id were deleted. Second, the date and time stamp 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.

The final list, now containing 51 entries per message was recoded to better suit a Neural Network by coding each of these entries individually. Since the original coding scheme has several options per entry (such as 4 different classifications 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. In the case of the number of message lines this led to four features, each of them having only two possible values: 1 (or `on`), indicating the feature is present; or 0 (`off`), indicating the feature is absent. Note that each *group of features* which resemble one entry in the original database always has one option chosen; that is, each group of features is mutually exclusive. The recoding resulted in 149 binary features in the new database. Appendix A lists all entries and how they are split into groups of features.

3 Autoassociative Neural Networks

Autoassociative Neural Networks (ANN) are special kinds of Neural Networks which are used to simulate (and explore) associative processes. Association in these types of Neural Networks is achieved through the interaction of a set of simple processing elements (called *units*), which are connected through *weighted connections*. These connections can be positive (or *excitatory*), zero (which indicates no correlation between the connected units), or negative (*inhibitory*). The value of these connections is learned through the *training* process of the ANN. During training, patterns are presented to the network and the weights are gradually adjusted in a way that the final pattern of connectivity matches all patterns being presented. One complete presentation of all patterns with which the network is trained is called one *epoch*; usually a network requires many such epochs to perform satisfactorily. The weights can therefore be seen as a distributed representation of the data.

The interpretation of the units on the other hand can be manifold, for example they can represent aspects of things (*features*) or they can symbolize certain actions or goals. Yet another possibility would be that single units represent hypothesis about certain properties of a model.

In a more formal way an ANN can be defined by the following properties:

- a *set of n units*, operating in parallel, each of them having
- an *activation rule* a_i, which leads to
- a *state of activation* $(a_1, \ldots, a_n)$ resulting in
- an *output vector* $(o_1, \cdots, o_i, \cdots, o_n)$, using
- an individual *output function* $(o_i = F_i(a_i))$ for each of the units. These output-values are transferred using
- a set of *weighted connections* $w_{i,j}$ between units. And
- a *learning rule* is used to modify the weights $w_{i,j}$ to learn properties of specific *training examples*.

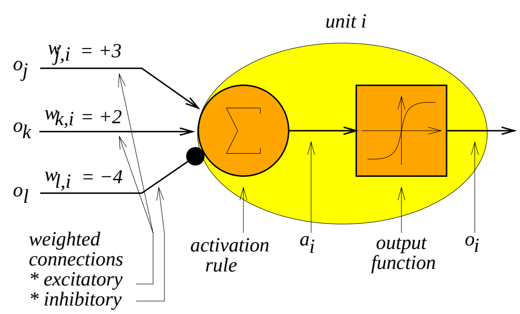

Fig. 1. An example of one unit of an Autoassociative Neural Net.

Figure 1 illustrates one unit of an ANN. In most cases the activation rule is nothing more than a weighted sum with an additional individual threshold θ_i:

$$a_i = \sum_{i=1}^{n} w_{i,j} o_j + \theta_i . \tag{1}$$

The purpose of the output function is mainly to ensure that the output value stays in a predetermined range. Usually the so called step function:

$$o_i^l(a_i) = \begin{cases} 1 & : \quad a_i \geq 0 \\ 0 & : \quad a_i < 0 \end{cases} \tag{2}$$

or a sigmoid:

$$o_i^s(a_i) = \frac{1}{1 + e^{\beta a_i}} \tag{3}$$

is used (see figure 2 for a graph of the functions). The latter has the advantage

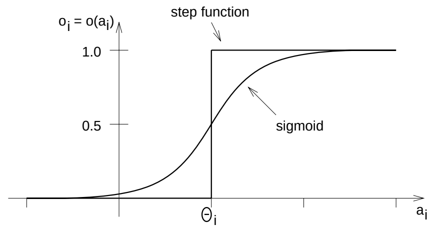

Fig. 2. The two most commonly used output function, the step function and a sigmoid.

of being differentiable which is a requirement for many learning algorithms since they compute the derivative of the error.

For the case of θ_i being 0 and the step activation function, the units will become active (or *fire*) if the sum of all excitatory stimuli is greater than the sum of all inhibitory stimuli. The threshold θ_i is only used to shift the zero point of the activation function. This threshold allows units to have an a–priori excitatory or inhibitory state which has to be overcome by external influences.

Figure 3 shows an example of an ANN with three units. An additional pseudo-unit is used to model the threshold values. It has a constant value of +1 and is connected via weighted connections to the normal units. Note that we do not include weights with value zero, e.g. there is an arc between the threshold unit and unit 2, but since the value of the corresponding weight is zero it is not shown. The usual convention of drawing excitatory weights as arrows and inhibitory weights as lines ending in a point is used here.

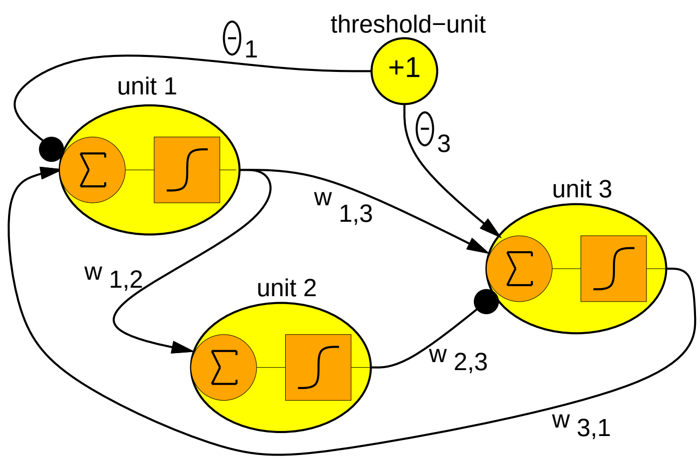

Fig. 3. An example for an ANN: Three units are connected to each other via excitatory (arrows) and inhibitory (lines ending in dots) connections. In addition a threshold unit with a constant value of +1 is shown. Note that connections with a zero-weight are not drawn.

To compute the state of activation of such a network, two methods were presented in (?). In the case of a *synchronous update*, the new activation value of all units is computed at once and then assigned to the outputs at the next point in time. Since this method can lead to oscillations in the output values, another method is often used, the *asynchronous update* (see (Hopfield, 1981)), in which a unit's activation is updated and propagated to its output immediately. The order in which the units are updated is random every time. This method of computation usually avoids oscillations.

Of course, if one would use this straightforward method of computing the state of activation for the network one might get stuck in a local minima. Since the state the computation is started from is very important and the units updated first have a major influence on the final outcome, a stochastic way of updating activations is normally used. An additional parameter, called *temperature* (T) is introduced and is used to cool the system down over time. Instead of using the step function to compute the output of a specific unit a function is used that introduces an undeterministic behaviour:

$$p_i = \frac{1}{1 + e^{\frac{\beta a_i}{T}}} \tag{4}$$

Figure 4 shows the probability distribution for two different settings of the parameter T. This equation computes the probability for unit i to *fire* (or have an output value of 1). Using this technique an *annealing schedule* defines the cooling

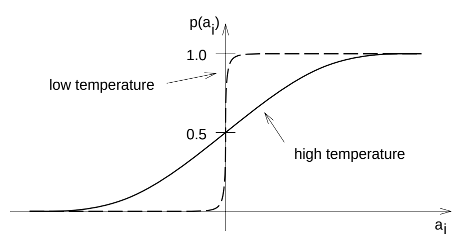

Fig. 4. The probability distribution for a unit to fire, depending on the activation value. The influence of the temperature is illustrated, a low temperature leads to an almost deterministic behaviour, at a high temperature the output is almost random for small activation values.

process. Usually it starts with a very high temperature where most activations are mainly chosen randomly and the temperature declines in an exponential manner until it reaches a point where the stochastic influence is almost non existent (the network *freezes*). The lower the temperature gets, the weaker the influence of the temperature gets, and the network behaves more and more deterministically and the output of the units is more likely to be equal to the value computed by the step function. This methodology can be proven to provide an optimal solution under certain circumstances.

The following example illustrates the use of ANNs.

3.1 The "animals in one room" problem

This is a famous computer-science problem. Assume there is a room with four possible inhabitants: dogs, cats, mice, and elephants:

– dogs: they chase cats (and since these are big dogs, they are stronger than cats too), don't care about mice and get stepped on by elephants.
– cats: they kill mice and get chased by dogs, but they are too fast for the elephants.
– mice: they frighten elephants and get killed by the cats, dogs just ignore them.
– elephants: they are frightened of mice and step on dogs, but not cats since they are too fast.

An appropriate network to model these facts would be one with four units, each representing the absence or presence of one the above species. Each of these units is connected to all of the other units with a weighted (uni–directional) connection, originating at one animal (A) and pointing to another animal (B). If this weight is zero, animal A doesn't threaten animal B; on the other hand if the weight is negative (or inhibitory) animal A threatens (steps-on, chases, kills) animal B. The case that the weight is positive (or excitatory) doesn't occur in this example, as this would resemble animal A liking animal B. In this example only threatening factors are considered[5]. Figure 5 shows the connections between the four units together with a threshold–unit. The excitatory weight from the

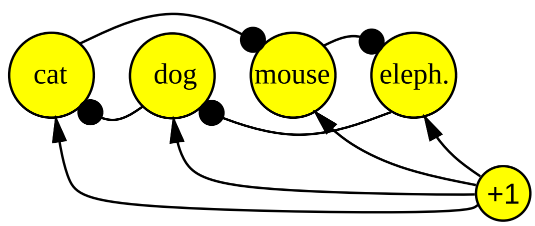

Fig. 5. The "animals in one room"–network: only connections with nonzero weights are shown, negative (inhibitory) weights are shown as lines ending in a point, arrows indicate a positive (excitatory) weight. The relationship between the different animals ("who threatens whom") is clearly shown. The excitatory weights originating at the threshold unit ensure the presence of at least one animal in the room.

threshold unit (having a constant output of +1) is used to ensure that at least one animal will be present. Otherwise a trivial solution (no animal in the room) would be enough to satisfy all constraints of the network. The final matrix of interconnections is shown in table 1. For larger networks this matrix is usually displayed graphically as shown in figure 6. This network can now start to cool

[5] A positive weight could be used to model for example that the cat would actually like to be in a room with a mouse, but this would lead to a more complex network.

Table 1. The interconnection matrix for the dog–cat–mouse–elephant network.

from/to	cat	dog	mouse	elephant
cat	—	0	-1	0
dog	-1	—	0	0
mouse	0	0	—	-1
elephant	0	-1	0	—
bias	$+0.5$	$+0.5$	$+0.5$	$+0.5$

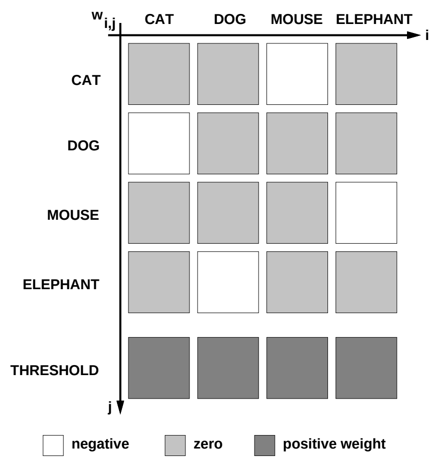

Fig. 6. A graphical way of displaying the weights of the example network. This kind of diagrams is often used for large weight–matrices.

down in a random state and the asynchronous update of the units executed until the temperature reaches zero and the activations of the units reach a constant state (the network *settles*), i.e. the calculation of the activation value of each unit from the sum of all weighted incoming activations compared with the thresold value does not change the state of the network anymore. Several iterations are usually necessary to reach this final state. The network will always settle in a state which represents a room occupied by a combination of animals who won't threaten each other (for example elephants and cats, but no dogs and mice).

If one of the units is forced to be `on` (or having an output value of 1), the energy landscape of the network will change and the only states it will settle in will have this particular unit to be `on` as well. This forcing (or *clamping*) of units restricts the feature space of solutions to a sub–space and therefore eliminates unwanted solutions.

3.2 The Hebbian Training Procedure

In the previous example the weights were assigned from common knowledge about the behaviour of animals. Here we show how weights can be calculated automatically from examples. The Hebbian learning rule (see (?)) trains the network through the presentation of examples and successive alterations of the weights.

The training examples are presented to the network one after the other, each unit is inspected and the weights leading to this unit (and therefore influencing its activation) are adjusted according to the following rules:

- When unit A and B are simultaneously excited (or correlated), increase the strength of the connection between them.
- When unit A and B are counter-correlated, decrease the strength of the connection between them.

This can be done by simply inspecting each of the units, comparing its activation and the activation of the units where the connection originates, and decreasing or increasing the weight accordingly.

In the previously used example one would use a database with several possible examples of animals in the room and use this to train the network. The network would pick up relationsships between animals and adjust its weights accordingly. For example, since mouse and cat will never appear together weights connecting the corresponding units would only decrease (and thus become inhibitory), never increase.

For more complex tasks a more sophisticated training algorithm is used, but it is still based on the described method. It employs a stochastic component and also makes use of the thresholds. In the following chapter the more advanced algorithm will be used.

4 Applying an ANN to 3000 Messages

The network used for the experiments presented in this paper, an autoassociative neural network, will be described in the following section.

4.1 The ANN

Since the data consisted of 149 features, each taking a value of either "0" or "1" after processing, the network has 149 binary units. This leads to $149 * 149 = 22,201$ weights and 149 thresholds to adjust during training. The idea of this type of network is to present each of the 3000 training patterns to the network and adjust the weights in a way which stores the information contained in each of the patterns. Each unit is connected via a directed arc with each other unit and thus allowing them to have an excitatory (positive weight) or inhibitory (negative weight) influence on each other individually. The pattern of connectivity (or *weight matrix*) will be explored in section 5.1.

The annealing schedule for pattern completion started at a temperature of $T = 500$ and declined exponentially in 100 steps to its final value of $T = 0$. During each step 20 asynchronous updates of all units were performed.

4.2 Training a Network with over 20,000 weights

Training this neural network obviously requires millions of computations. In this case the network consists of $22,350$ parameters ($22,201$ weights and 149 threshold values) to adjust and to update each of them 149 activations have to be computed, each of them requiring again 150 weighted summations. This has to be done for each of the $3,000$ patterns in the database, which leads to a required computation of over 65 million weighted summations (or *connections*) plus roughly the same amount of compare- and update-operations per epoch. Almost 100 hours of CPU–time were spent before the error-rate of the network started to settle on a plateau (see figure 7) after five days. Figure 8 shows a

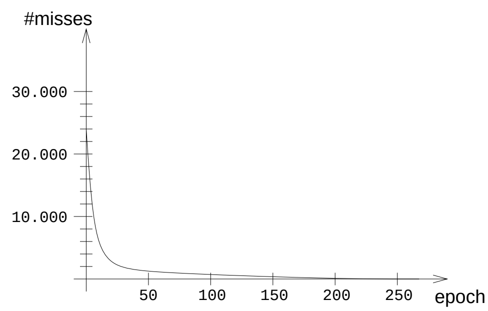

Fig. 7. The decreasing number of misses vs. training time is shown in this graph. After five days of training, when the error finally reached a plateau, training was stopped.

weight diagram. Each of the weights is presented as a small rectangle. Its colour indicates the value of the corresponding weight: black means a negative value, grey equals zero and white indicates a positive weight.

5 Results drawn from the trained ANN

5.1 Interpreting the weight matrix

Looking onto the weights (figure 8) one interesting property comes to mind: using the $149 * 150 = 22,350$ weights the network is able to recall the $3,000$ messages almost perfectly. $3,000 * 149 = 447,000$ `on/off`–informations (or *bits*) are stored in approximately $67,000$ bits (each weight stores about 3 bit), a compression factor of close to one order of magnitude! In addition the access to common features of all messages is much easier using the Neural Network compared to a global search in the whole database.

A closer look at the weight matrix shown in figure 8 gives some interesting insights. Here only a few examples are listed:

- the dark squares along the diagonal line show that all features of one group are mutually exclusive, only one of them can have the value 1 at a time. This

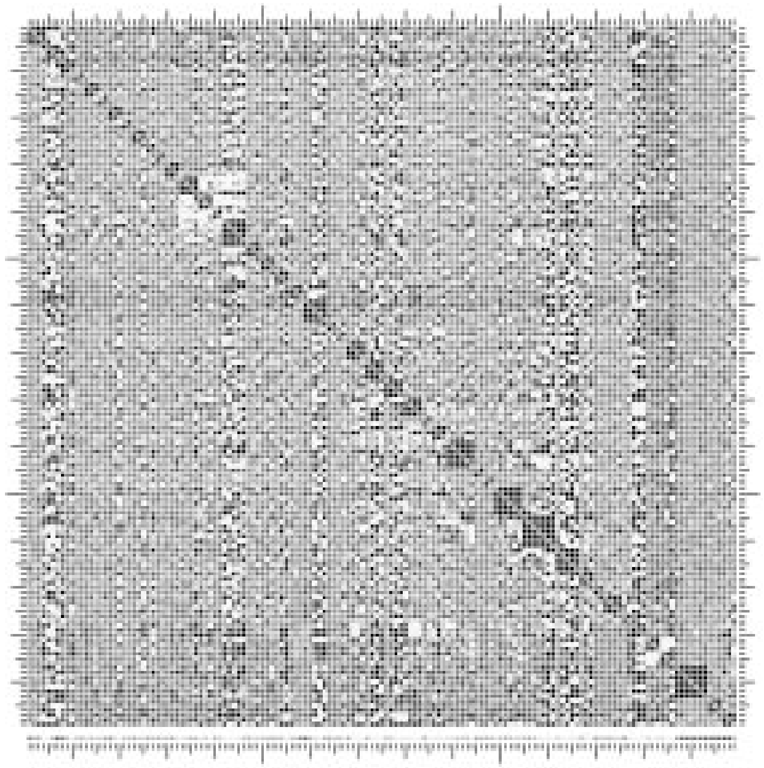

Fig. 8. The matrix of $149 * 149$ weights. Dark grey means a negative value, lighter grey values indicate positive weights. The bottom row of points represents the threshold values of each unit.

results in strong inhibitory connections from one feature to all others in one group and can be seen as a block of black points along the diagonal with a size according to the number of features in this group.

– some excitatory connections (shown by white squares at the appropriate position):
  - in column 1, corresponding with the feature MSGLINES-A (1–10 lines of original text, see also Appendix A) and row 13 (OPINION-A = no opinion is stated) an almost white square indicates a strong excitatory connection from unit 1 to unit 13. 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.
  - column 1, row 43: short messages(1) request information (43)
  - column 33, row 1: unformatted messages (33) are short (1)

– and another inhibitory connection (black square):
  • if message contains artistic icon(s) (54, 55) it is **not** short (1)

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 modelled 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 this technique was used to define certain properties of a message and let the network determine which other features are correlated.

This leads to the creation of *typical examples* for specific features being `on`, discussed in the next section.

5.2 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 work described in ((Coyne and Yokozawa, 1992)) but in the case of the ProjectH–Network, there are several states the network can settle in. This is mainly due to the fact that the input data is not free from noise and not all of the units are going to be strongly correlated to other units (or correlated at all). This would be analogous to a very rough and elastic energy landscape. Elastic because every move is going to change activations of states and those changes are again changing the energy landscape. Therefore the random update of the states will not always result in same final state for the network. Instead one would expect the unaffected states 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 2 shows the frequency of 1's occurring for some 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 might not be 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 formating. 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, each with an indication of how often the network settled in a state which had this particular

Table 2. The frequency of feature activations for feature no 29 (message contains humour) clamped. The columns show the number of the feature, its description and the frequency of `on`–activations.

no.	description	#1s
1	LINES-A : 1-10 lines of real mes.	40%
2	LINES-B : 11-25 lines of real mes.	50%
3	LINES-C : 26-100 lines of real mes.	10%
4	LINES-D : > lines 100 of real mes.	0%
5	SUBJECT-A : no subject line	5%
6	SUBJECT-B : subject line is appropriate	95%
7	SUBJECT-C : subject line is inappropriate	0%
	...	
21	QUESTION-A : no question/request contained	70%
22	QUESTION-B : mes. contains question/request	30%
	...	
28	HUMOUR-A : no humour contained	0%
29	HUMOUR-B : mes. contais humour	100%
	...	
33	FORMAT-A : unformatted	25%
34	FORMAT-B : minimal formatted	35%
35	FORMAT-C : mostly formatted	35%
36	FORMAT-D : overformatted	5%
	...	
56	GENDER1-A : can't tell	0%
57	GENDER1-B : female	20%
58	GENDER1-C : male	80%
	...	
112	FLAME2-A : no abusive language	100%
113	FLAME2-B : abusive lang. about content only	0%
114	FLAME2-C : abusive lang. about person	0%
115	FLAME2-D : abusive lang. about general. others	0%
116	FLAME2-E : mixture	0%
	...	

state being `on`. This information can be used to produced typicality sets as shown in the next section.

5.3 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 (Θ) can be used to choose features for this set. If every two members of the typicality set are required to have appeared at least 60% of all times together, the following computes Θ:

$$\Theta = 100\% - n(100\% - p) = (1 - n)100\% - np \tag{5}$$

with $n = 2$ and $p = 60\%$. This leads to a threshold Θ of 80%. If a higher likelihood for the appearance of features together (p) or a higher number of features occurring together (n) is desired, the threshold of selection (Θ) will increase.

The example from the previous section is again used to show its typicality set (see table 3).

Table 3. The typicality set for feature 29 (message contains humour)

no.	label	description
6	SUBJECT-B	subject line is appropriate
9	NOISE-B	regular msg
18	APOLOGY-A	no apology
26	CHALLENGE-A	no challenge/bet/dare
38	STYLE1-B	regular capitalization
53	ARTICON-A	no artistic icons
58	GENDER1-C	male
68	QUOTE1-A	no quoted text from this list
72	QUOTE2-A	no CMC text quoted from outside list
95	COALIT2-A	no first person plural
98	COALIT3-B	addresses other person
112	FLAME2-A	no abusive language
117	FLAME3-A	no intention to prevent/calm tension
120	STATUS-A	no identification of status
126	SIGNAT2-A	no ending quotation
145	EVENING	6pm–12am

This table shows which features seem to be highly correleated with feature 29. But so far there is no information about the quality of the list. It could well be 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 most of the time just behave randomly and it was neccesary to score the *sensitivity* of each feature.

5.4 Scoring Features and Sensitivity

Of course, some of the features in the typicality set might not be as interesting as others. Some features 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 1s over all typicality sets for single features (this leads to 149 typicality sets, one for each feature). For the further analysis the percentage of 1s in the case of a clamped feature was compressed into 5 classes:

$\oplus$ between 80% to 100% of 1s in one experiment
$+$ 60% to 80%
$\bullet$ 40% to 60%
$-$ 20% to 40%, and
$\ominus$ 0% to 20% 1s (or 100% to 80% 0s in the experiment)

Taking again the example where feature 29 is clamped, table 4 shows a few of those classifications.

Table 4. The frequency of feature activations for feature 29 (message contains humour) clamped. The last column shows the classification.

no.	label	description	#1s	class
1	LINES-A	1-10 lines of real mes.	40%	$\bullet$
2	LINES-B	11-25 lines of real mes.	50%	$\bullet$
3	LINES-C	26-100 lines of real mes.	10%	$\ominus$
4	LINES-D	> lines 100 of real mes.	0%	$\ominus$
...				
21	QUESTION-A	no question/request contained	70%	$+$
22	QUESTION-B	mes. contains question/request	30%	$-$
...				
28	HUMOUR-A	no humour contained	0%	$\ominus$
29	HUMOUR-B	mes. contais humour	100%	$\oplus$
...				
33	FORMAT-A	unformatted	25%	$-$
34	FORMAT-B	minimal formatted	35%	$-$
35	FORMAT-C	mostly formatted	35%	$-$
36	FORMAT-D	overformatted	5%	$\ominus$
...				
56	GENDER1-A	can't tell	0%	$\ominus$
57	GENDER1-B	female	20%	$-$
58	GENDER1-C	male	80%	$\oplus$
...				

Taking all 149 typicality sets it is easy to compute 5 global values for each feature, the frequency with which the feature was covered by that specific class over all experiments. Table 5 shows a few examples from the table of all features.

With these five numbers a number can be computed to measure what sensitivity of a feature really means. If for example the percentage of $\oplus$s for one feature is a perfect 100%, this specific feature is always `on`, but since it is never `off` it does not really help to *distinguish* between different classes of messages. It does however tell us about a typical message in the whole set database. On the other hand, a feature having 50% $\oplus$ and 50% $\ominus$ would be much better suited to group messages in the database, in fact this is the best case one could imagine. Somewhere in between are features with unbalanced percentages of $\oplus$ and $\ominus$.

To generate a unifiying score for all features the following four heurisitcs were chosen:

1. A sensitive feature has at least one $\oplus$ (apart from the case where that feature was clamped) and one $\ominus$.
2. A feature is more sensitive than another one if the number of $\oplus$ is better balanced to the number of $\ominus$.
3. A smaller number of $+$, $\bullet$ and $-$ indicates a sensitive feature.
4. An insensitive feature has either no $\ominus$ or no $\oplus$ and a high number of $+$, $\bullet$ and $-$.

These heuristics lead to the following way of computing the *sensitivity* s_i of a feature i:

$$s_i = \begin{cases} \frac{(N(\oplus)+N(\ominus))*(min\{N(\oplus),N(\ominus)\})}{max\{N(\oplus),N(\ominus)\}} & : \quad N(\ominus) \neq 0 \wedge N(\oplus) \neq 0 \\ \frac{N(\ominus)+N(\oplus)}{N(-)+N(\bullet)+N(+)} & : \quad N(\ominus) = 0 \vee N(\oplus) = 0 \end{cases} \tag{6}$$

with $N(\cdot)$ returning the frequency of the symbol passed as an argument. The *sensitivity–score* $s_i \in [-100, +100]$ and $s_i = +100$ in the best case, where both $\oplus$ and $\ominus$ occur 50% of all times (and therefore $-$, $\bullet$ and $+$ occur not at all) and the worst case ($s_i = -100$) having no $\oplus$ or $\ominus$ but only $-$, $\bullet$ and $+$. Figure 5 shows a few examples from the 149 features.

Table 5. A few examples of scored features. For each feature the percentage of times it got classified as being in a specific class is shown and the final score resulting from these classifications is listed in the last column.

no.	label	description	$\ominus$	$-$	$\bullet$	$+$	$\oplus$	score
...								
28	HUMOUR-A	no humour	1%	12%	32%	42%	12%	+1
29	HUMOUR-B	contains humour	13%	42%	32%	11%	0%	−23
...								
33	FORMAT-A	unformatted	53%	39%	6%	0%	0%	−66
34	FORMAT-B	minimal formatted	34%	49%	13%	2%	0%	−49
35	FORMAT-C	mostly formatted	17%	42%	30%	8%	0%	−28
36	FORMAT-D	overformatted	90%	9%	0%	0%	0%	−93
...								
40	STYLE2-A	no colloquial spelling	4%	14%	22%	43%	16%	+5
41	STYLE2-B	contains colloquial spelling	17%	43%	22%	14%	2%	+2
...								
56	GENDER1-A	can't tell	100%	0%	0%	0%	0%	−100
57	GENDER1-B	female	75%	22%	2%	0%	0%	−82
58	GENDER1-C	male	2%	0%	3%	38%	56%	+2
...								

This table shows how some features are not very sensitive towards the activtions (caused by clamping) of others. A good example would be the format of the messages (FORMAT-A to FORMAT-D), all features have a negative sensitivity score because none of them appears in another typicality set besides its own. In contrast the STYLE2–entry has for both features positive scores. Looking at the table it becomes clear why. STYLE2-A appears in 5% of all typicality sets and is completely absent in 4% of all messages. Almost the same goes for STYLE2-B, it is absent for 17% and included in the typicality set for 2% of all cases. As expected, none of the features reached the perfect score of +100.

6 Typicality in CMC

The previous section described how typicality sets for single features can be generated by an Autoassociative Neural Network. For an analysis of typicality in CMC, and especially an investigation of threads and their characteristics, some typicality sets are more interesting than others.

To analyse the nature of threads the comparison of a message which starts or continues a thread vs. a message which ends a thread is interesting. Figure 9 shows a reference–tree to illustrate the used terms *reference–width*, *reference–height* and *reference–depth*. 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.

To characterize a "good" vs. a "bad" message in the sense of participation in a thread the variable reference–width was used. A message is called "good" if it as at least referenced by one other message, it *participates* in a thread. In contrast a "bad" message is not referenced at all, 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. They are shown in tables 6 and 7. 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. To create the final typical "good" and "bad" message, features appearing in both typicality sets will be deleted from both sets and features with a too low sensitivity score will be discarded too. This leads to tables 8 and 9 and finally enables us to extract some properties of the messages in the database:

– a "good" message has medium length (2) and an appropriate subject line (6).

Table 6. The typicality set of a "referenced" message. Clamped feature: 135 (MSGWIDTH-B: 1–2 references to this message)

no.	feature description	sensitivity–score
2	11-25 lines of original text	1
6	subject line is appropriate	1
12	contains verbal selfdisclosure	1
17	contains statement of a fact	1
21	no question/request	1
26	no challenge/bet/dare	1
38	regular capitalization	2
47	no emoticons	2
50	no punct device to express emotion	3
53	no artistic icons	1
58	male	2
60	identifies gender via name/signature	3
68	no quoted text from this list	2
72	no CMC text quoted from outside list	2
98	addresses other person	1
112	no abusive language	2
120	no identification of status	1
126	no ending quotation	1

Table 7. The typicality set of a "nonreferenced" message. Clamped feature: 134 (MSGWIDTH-A: no references to this message)

no.	feature	sensitivity–score
12	contains verbal selfdisclosure	1
26	no challenge/bet/dare	1
28	no humour	1
38	regular capitalization	2
53	no artistic icons	1
68	no quoted text from this list	2
80	no prev msg referenced by this msg	4
87	new topic, no ref to prev discussion	18
95	no first person plural	1
112	no abusive language	2
117	no intention to prevent/calm tension	2
120	no identification of status	1
126	no ending quotation	1
128	no prev msg referenced by this msg	2
131	no references after this msg	9

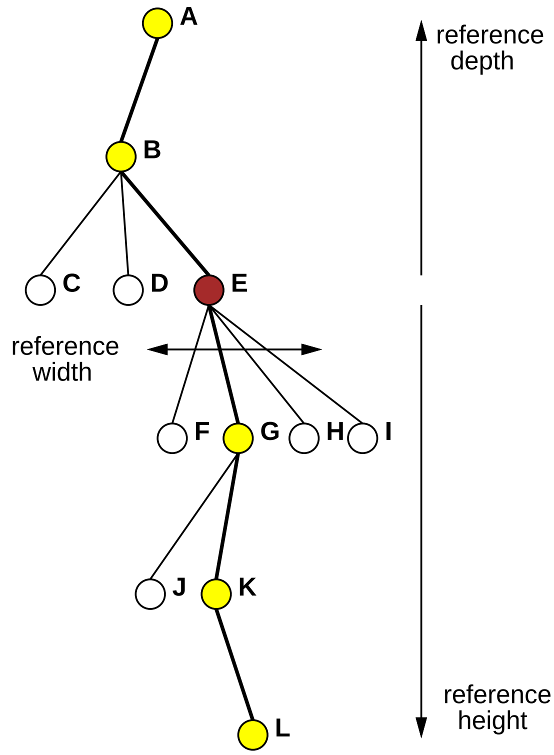

Fig. 9. A reference–tree, illustrating the terminology. A thread starts with message A and the last message participating is L. Message E is being reference directly by four messages (the *reference–width*), references itself a sequence of two message (*reference–depth*) and is reference by a sequence of three messages (*reference–height*)

- a statement of a fact (17) also enhances the chances of being followed–up.
- if during an already ongoing thread one introduces a completely new topic (87), the chances of getting a response are slim. This point seems to be a very strong one, regarding the high sensitivity score of that specific feature.
- interesting also is that a message which does not reference seems likely not to be referenced. But the sensitivity score of this feature is reasonable low, which makes sense, otherwise threads would never start. But this discovery indicates that the start of a thread is not an easy task. Being followed–up when one already participates in a thread is much easier.

7 Conclusions

We have described an approach to use Autoassociative Neural Networks to explore typicality in computer mediated discussions. We showed how to train an ANN and how the final weight matrix can be used to extract relationships

Table 8. Typical distinguishing features of a referenced (or "good") message

no.	feature description	sensitivity–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

Table 9. Typical features of a nonreferenced (or "bad") message

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

between variables. We then used to Neural Network to extract typicality sets for specified features and showed how messages which support threads ("good" messages) can be distinguished from those messages not participating in a thread ("bad" messages).

This sort of 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 would thus only be used to discover feature–groups that are correlated and further statistics 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 strongly correlated while other features are only loosely or not at all connected. In contrast to the example used by Coyne et al. (1993), noise from coder–errors as well as differences on opinionated variables (as described by Rafaeli and Sudweeks (1996)) result in a database which is not as well structured as artificial ones.

The possibilities using an ANN are far from being exhausted and several features are well worth exploring, STYLE2 for example, which has both high sensitivity scores for all features of the group and a fairly unbalanced frequency distribution (see appendix A for a listing). This would be another way of exploring threads. But also an analysis about the quality of interactivity could be performed, by using feature–group DEPEND3, which describes the manner in which previous messages are referenced. Yet another example is GENDER3 which is also a feature group with a good distribution and high sensitivity–

scores. GENDER3 codes the fact that gender–identification is an issue. The same approach can also be used to find features that are a typical for messages. Features with a very low sensitivity–score and a typical value of 0 have a strong negative correlation with almost every other feature. This would lead to an "anti–message" with an a typicality set.

Obviously only a very specific kind of Neural Network was used for this analysis, more architectures are being published every day. The ANN was chosen because the autoassociative structure supports the emergence of examples; if the main focus of analysis were on only a few variables, a feedforward architecture would also be feasible. An approach using feedforward Neural Networks would create a network to *classify* examples rather than create an environment for emerging examples. If a Localized Receptive Field Network (Moody and Darken, 1989) were used the prototypes represent typical examples for each class and the radii and weights of those reference vectors are indicators for the value and generality of the example.

The approach we presented in this paper is obviously capable of extracting a form of relationships between features, but the ANN–aproach also helped to verify tentative hypotheses pertaining to computer–mediated communication as most results reported by the Neural Network did "make sense".

8 Acknowledgements

This work was supported by a University of Sydney Research Grant (URC).

References

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

Collins, A. M. and Loftus, E. F.: 1975, A spreading-activation theory of semantic processing, *Psychological Review*, **82**, 407-28.

Coyne, R. D., Newton, S. and Sudweeks, F.: 1993, Modelling 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.

Everitt, B.: 1974, *Cluster Analysis*, Wiley, New York.

Hertz, J., Krogh, A. and Palmer, R. G.: 1991, *Introduction to the Theory of Neural Computation*, Addison-Wesley, Redwood City, CA.

Hopfield, J. J.: 1981, Neural networks and physical systems with emergent collective computational abilities, *Proceedings of the National Academy of Sciences*, USA, 3088-3092.

Kahneman, D. and Tversky, A.: 1973, On the psychology of prediction, *Psychological Review*, **80**, 237-51.

Konstan, J. A.: 1994, Unix to the rescue: using Unix in communications research, Proceedings Usenix Conference (to appear).

Mehra, P. and Wah, B. W. (eds): 1992, *Artificial Neural Networks*, IEEE Computer Society Press.

Moody, J., Darken, C. J.: 1989, Fast Learning in Networks of Locally–Tuned Processing Units *Neural Computation* **1**, 281-294.

Newton, S.: 1992, On the relevance and treatment of categories in AI in design, *in* J. S. Gero (ed.), *Artificial Intelligence in Design '92*, Kluwer, Dordrecht, pp. 861-882.

Pulman, S. G.: 1983, *Word Meaning and Belief*, Croom Held, London.

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

Rafaeli, 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.

Rafaeli, S. and Sudweeks, F.: 1996, Interactivity on the Net, *in* Sudweeks, F. McLaughlin, M. and Rafaeli, S. (eds), *Network and Netplay: Virtual Groups on the Internet*, AAAI/MIT Press (to appear).

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

Rogers, E. M. and Rafaeli, S.: 1985, Computers and communication, *in* B. D. Ruben (ed.), *Information and Behavior*, Vol. 1, Transaction Books, New Brunswick, NJ, pp/ 135–155.

Rosch, E.: 1978, Principles of categorization, *in* E. Rosch and B. B. Lloyd (eds), *Cognition and Categorization*, Lawrence Erlbaum, Hillsdale, New Jersey, pp. 27–48.

Rosenman, M. A. and Sudweeks, F.: 1995, Categorisation and prototypes in design, *in* Slezak, P., Caelli, T. and Clarke, R. (eds), *Perspectives on Cognitive Science: Theories, Experiments and Foundations*, Albex, Norwood, NJ, pp. 189-212.

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.

Smith, E. E. and Medin, D. L.: 1981, *Categories and Concepts*, Harvard University Press, Cambridge, Massachusetts.

Sudweeks, F. and Rafaeli, 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).

Tversky, A.: 1977, Features of similarity, *Psychological Review*, **84**, 327–52.

APPENDIX A: The List of final Features

no.	old label	new no.	new label	meaning	frequency	score
1	CODERID			deleted		
2	LISTID			recoded, see no. 52		
3	MSGNUM			deleted		
4	AUTHORID			deleted		
5	MSGTIME			recoded, see no. 51		
6	MSGDATE			recoded, see no. 50		
7	MSGLINES	1	LINES-A	1-10 lines of original text	66.4%	-53%
		2	LINES-B	11-25 lines of original text	22.2%	+1%
		3	LINES-C	26-100 lines of original text	8.7%	-83%
		4	LINES-D	>lines 100 of original text	2.4%	-97%
8	SUBJECT	5	SUBJECT-A	subject line is inappropriate	15.1%	-92%
		6	SUBJECT-B	subject line is appropriate	81.8%	+1%
		7	SUBJECT-C	no subject line	2.4%	-99%
9	NOISE	8	NOISE-A	msg not intended for this list	5.6%	-76%
		9	NOISE-B	regular msg	89.4%	+2%
		10	NOISE-C	msg intended for list but irregular	4.8%	-100%
10	FIRSTPER	11	FIRSTPER-A	no verbal selfdisclosure	64.6%	-63%
		12	FIRSTPER-B	contains verbal selfdisclosure	35.1%	+1%
11	OPINION	13	OPINION-A	no opinion is stated	50.9%	-36%
		14	OPINION-B	opinion stated, not main item	30.2%	+1%
		15	OPINION-C	opinion stated, is main item	18.5%	-70%
12	FACT	16	FACT-A	no statement of a fact	45.1%	-84%
		17	FACT-B	contains statement of a fact	54.6%	+1%
13	APOLOGY	18	APOLOGY-A	no apology	93.3%	+1%
		19	APOLOGY-B	contains mild apology	4.5%	-99%
		20	APOLOGY-C	contains clear apology	1.9%	-98%
14	QUESTION	21	QUESTION-A	no question/request	71.9%	+1%
		22	QUESTION-B	contains question/request	27.8%	-66%
15	ACTION	23	ACTION-A	no call for action	90.7%	-39%
		24	ACTION-B	call for action, not main item	6.4%	+3%
		25	ACTION-C	call for action, is main item	2.6%	-72%
16	CHALLENGE	26	CHALLENGE-A	no challenge/bet/dare	95.9%	+1%
		27	CHALLENGE-B	contains challenge/bet/dare	3.8%	-82%
17	HUMOR	28	HUMOUR-A	no humour	80.5%	+1%
		29	HUMOUR-B	contains humour	19.2%	-23%
18	METACOMM	30	METACOMM-A	no metacommunication	85.2%	+1%
		31	METACOMM-B	metacommunication, not main item	8.6%	-91%
		32	METACOMM-C	metacommunication, it is main item	5.9%	-74%
19	FORMAT	33	FORMAT-A	unformatted	13.3%	-66%
		34	FORMAT-B	minimal formatted	64.5%	-49%
		35	FORMAT-C	mostly formatted	19.6%	-28%
		36	FORMAT-D	overformatted	2.2%	-93%

no.	old label	new no.	new label	meaning	frequency	score
20	STYLE1	37	STYLE1-A	minimal or no capitalization	20.4%	-96%
		38	STYLE1-B	regular capitalization	77.2%	+2%
		39	STYLE1-C	mostly or all capitalization	2.0%	-100%
21	STYLE2	40	STYLE2-A	no colloquial spelling	89.0%	+5%
		41	STYLE2-B	contains colloquial spelling	10.6%	+2%
22	NATURE	42	NATURE-A	provides information	39.4%	+2%
		43	NATURE-B	requests information	16.4%	-95%
		44	NATURE-C	persuasive	3.5%	-89%
		45	NATURE-D	opinionated	18.5%	-89%
		46	NATURE-E	mixed style	21.1%	-71%
23	EMOTICON	47	EMOTICON-A	no emoticons	88.1%	+2%
		48	EMOTICON-B	contains 1 emoticon	8.6%	-53%
		49	EMOTICON-C	contains>1 emoticon	2.9%	-99%
24	EMODEVICE	50	EMODEVICE-A	no punct device to express emotion	88.0%	+3%
		51	EMODEVICE-B	contains 1 punctuation device	7.3%	-53%
		52	EMODEVICE-C	contains>1 punctuation device	4.4%	-84%
25	ARTICON	53	ARTICON-A	no artistic icons	94.9%	+1%
		54	ARTICON-B	contains 1 artistic icon	2.5%	-99%
		55	ARTICON-C	contains>1 artistic icon	2.2%	-99%
26	GENDER1	56	GENDER1-A	can't tell	13.6%	-100%
		57	GENDER1-B	female	14.3%	-82%
		58	GENDER1-C	male	71.9%	+2%
27	GENDER2	59	GENDER2-A	does not identify gender	19.4%	-90%
		60	GENDER2-B	identifies gender via name/signature	75.6%	+3%
		61	GENDER2-C	identifies gender directly	1.1%	-94%
		62	GENDER2-D	identifies gender indirectly	2.2%	-98%
		63	GENDER2-E	mixture	1.5%	-95%
28	GENDER3	64	GENDER3-A	no gend spec terms regarding others	79.1%	-55%
		65	GENDER3-B	gender spec terms regarding others	20.6%	+1%
29	GENDER4	66	GENDER4-A	no gender identification issue	94.8%	+4%
		67	GENDER4-B	gender identification is an issue	4.9%	+2%
30	QUOTE1	68	QUOTE1-A	no quoted text from this list	70.4%	+2%
		69	QUOTE1-B	1-10 lines of quoted text	22.2%	-97%
		70	QUOTE1-C	11-25 lines of quoted text	5.4%	-100%
		71	QUOTE1-D	>26 lines of quoted text	1.6%	-100%
31	QUOTE2	72	QUOTE2-A	no CMC text quoted from outside list	95.8%	+2%
		73	QUOTE2-B	1-10 lines of CMC text quoted	1.2%	-97%
		74	QUOTE2-C	11-25 lines of CMC text quoted	1.0%	-97%
		75	QUOTE2-D	>26 lines of CMC text quoted	1.6%	-89%
32	QUOTE3	76	QUOTE3-A	no non-CMC text quoted	92.5%	+2%
		77	QUOTE3-B	1-10 lines of non-CMC text quoted	3.9%	-61%
		78	QUOTE3-C	11-25 lines of non-CMC text quoted	1.8%	-97%
		79	QUOTE3-D	>26 lines of non-CMC text quoted	1.4%	-92%
33	DEPEND1	80	DEPEND1-A	no prev msg referenced by this msg	30.9%	+4%
		81	DEPEND1-B	1 prev msg is referenced	52.0%	-74%
		82	DEPEND1-C	>1 prev msg is referenced	5.9%	-100%
		83	DEPEND1-D	a sequence of msgs is referenced	10.7%	-92%
34	DEPEND2			recoded, see no. 47—49		

no.	old label	new no.	new label	meaning	frequency	score
35	DEPEND3	84	DEPEND3-A	no ref to manner of a prev ref	88.7%	+4%
		85	DEPEND3-B	reference to the manner of prev ref	11.0%	+2%
36	DEPEND4	86	DEPEND4-A	clearly part of an ongoing thread	69.2%	+2%
		87	DEPEND4-B	new topic, no ref to prev discussion	23.3%	+18%
		88	DEPEND4-C	new topic, ref to prev discussion	7.1%	-96%
37	COALIT1	89	COALIT1-A	strong agreement with prev msg	6.5%	-89%
		90	COALIT1-B	mild agreement	8.4%	-78%
		91	COALIT1-C	no indication, neutral	67.3%	+2%
		92	COALIT1-D	both dis- and agreement	4.7%	-95%
		93	COALIT1-E	mild disagreement	8.5%	-97%
		94	COALIT1-F	strong disagreement	4.2%	-98%
38	COALIT2	95	COALIT2-A	no first person plural	89.9%	+1%
		96	COALIT2-B	contains first person plural	9.8%	-69%
39	COALIT3	97	COALIT3-A	does not address other persons	64.2%	-67%
		98	COALIT3-B	addresses other person	35.5%	+1%
40	EXTCOAL	99	EXTCOAL-A	strong agreement (outside list)	6.2%	-14%
		100	EXTCOAL-B	mild agreement	3.7%	-97%
		101	EXTCOAL-C	no indication, neutral	81.8%	-80%
		102	EXTCOAL-D	both dis- and agreement	2.3%	-91%
		103	EXTCOAL-E	mild disagreement	3.0%	-92%
		104	EXTCOAL-F	strong disagreement	2.5%	-90%
41	FLAME1	105	FLAME1-A	neutral or no opinion	57.4%	-18%
		106	FLAME1-B	friendly opinion	27.1%	-35%
		107	FLAME1-C	diverging opinion	6.5%	-100%
		108	FLAME1-D	disagreeing	4.1%	-100%
		109	FLAME1-E	tension	2.2%	-91%
		110	FLAME1-F	antagonistic	1.6%	-97%
		111	FLAME1-G	hostile	0.6%	-100%
42	FLAME2	112	FLAME2-A	no abusive language	95.9%	+2%
		113	FLAME2-B	abusive lang about content only	1.1%	-98%
		114	FLAME2-C	abusive lang about person	0.9%	-97%
		115	FLAME2-D	abusive lang about self/general	0.9%	-100%
		116	FLAME2-E	mixture	0.7%	-100%
43	FLAME3	117	FLAME3-A	no intention to prevent/calm tension	95.6%	+2%
		118	FLAME3-B	tries to prevent tension	1.8%	-99%
		119	FLAME3-C	tries to calm ongoing tension	2.2%	-88%
44	STATUS	120	STATUS-A	no identification of status	87.9%	+1%
		121	STATUS-B	status is identified	11.9%	-99%
45	SIGNAT1	122	SIGNAT1-A	no signature	29.1%	+2%
		123	SIGNAT1-B	simple signature	49.7%	-25%
		124	SIGNAT1-C	complex signature	15.6%	-98%
		125	SIGNAT1-D	artistic signature	5.3%	-99%
46	SIGNAT2	126	SIGNAT2-A	no ending quotation	87.7%	+1%
		127	SIGNAT2-B	contains ending quotation	12.0%	-95%
47	PREDMSG	128	PREDMSG-A	no prev msg referenced by this msg	32.7%	+2%
		129	PREDMSG-B	1-2 prev msgs referenced	44.5%	-92%
		130	PREDMSG-C	>2 prev msgs referenced	22.7%	-75%

no.	old label	new no.	new label	meaning	frequency	score
48	SUCCMSG	131	SUCCMSG-A	no references after this msg	61.4%	+9%
		132	SUCCMSG-B	1-2 references after this msg	25.8%	-21%
		133	SUCCMSG-C	>2 references after this msg	12.7%	-93%
49	MSGWIDTH	134	MSGWIDTH-A	no msgs are referencing this msg	61.4%	+9%
		135	MSGWIDTH-B	1-2 references to this msg	36.0%	+2%
		136	MSGWIDTH-C	>2 references to this msg	2.5%	-91%
50	WEEKDAY	137	MONDAY		8.1%	-97%
		138	TUESDAY		9.7%	-97%
		139	WEDNESDAY		19.0%	-42%
		140	THURSDAY		19.7%	-75%
		141	FRIDAY		14.4%	-100%
		142	SATURDAY		13.9%	-96%
		143	SUNDAY		15.0%	-77%
51	WORKTIME	144	WORKTIME	8am–6pm	20.6%	-98%
		145	EVENING	6pm–12am	50.7%	+2%
		146	NIGHT	12am–8am	28.7%	-95%
52	LIST-ID	147	COMPUSERVE		30.0%	-79%
		148	BITNET		36.6%	+1%
		149	USENET		33.3%	+10%