Semiotics and Agents for Integrating and Navigating through Multimedia Representations of Concepts

Dan W. Joyce, Paul H. Lewis, Robert H. Tansley, Mark. R. Dobie, Wendy Hall

Multimedia Research Group, Dept. Electronics and Computer Science
University of Southampton, Southampton, Hampshire, England, S017 1BJ

ABSTRACT

The purpose of this paper is two-fold. We begin by exploring the emerging trend to view multimedia information in terms of low-level and high-level components; the former being feature-based and the latter the "semantics" intrinsic to what is portrayed by the media object. Traditionally, this has been viewed by employing analogies with generative linguistics (e.g. compositional semantics). Recently, a new perspective based on the semiotic tradition has been alluded to in several papers. We believe this to be a more appropriate approach. From this, we propose an approach for tackling this problem which uses an associative data structure expressing authored information together with intelligent agents acting autonomously over this structure. We then show how neural networks can be used to implement such agents. The agents act as "vehicles" for bridging the gap between multimedia semantics and concrete expressions of high-level knowledge, but we suggest that traditional neural network techniques for classification are not architecturally adequate.

Keywords: content-based retrieval, semiotics, neural networks, intelligent agents

1. INTRODUCTION

Multimedia content-based retrieval and navigation (CBR/N) relies upon relatively low level feature extraction techniques to provide representations of the media objects. For example, an image might be represented in a retrieval database by features representing texture, colour distributions etc. More recently, the annotation of media objects with symbolic metadata has provided an alternative, "semantic feature" based description. In our system, MAVIS2[1] this high-level ontology (an account of *what there is* in a domain) is captured by means of a restricted conceptual (associative or semantic) network. Integrating both high-level metadata and low-level media feature information is a difficult problem – the correspondence between the semantics and (for example) visual appearance of media objects being particularly difficult to represent algorithmically. A number of workers have similarly recognised the inherent problem of associating (or grounding) high-level knowledge in low-level (feature-based) data[2] and more generally in cognitive science and artificial intelligence as the symbol grounding problem.[3]

2. INTRODUCTION TO MAVIS2

Figure 1 depicts the data structure at the root of the MAVIS2 system. The design intentions for MAVIS2 were to allow the user to navigate and retrieve from multimedia information spaces using traditional, content-based methods or, using the ontology described by the conceptual layer or a combination of these. Traditionally, text-based systems enable navigation by keywords, and the associative network of links between keywords is taken as a navigational ontology. Likewise, conventional content-based retrieval is feature similarity based only. The novelty in MAVIS2 is the integration of both levels, the higher knowledge-based level with lower-level feature-based methods. This paper focuses on intelligent agents that facilitate the process of navigation and retrieval between and within these two distinct levels. Essentially, the agents provide fast-track connections between classes of low-level data and the symbolic representations of classes to which they are deemed to belong. This has one significant engineering efficiency in terms of speed of search. The agent's learn (by training) associations between the levels and respond quicker than if, for each query, the whole of the mediating layers (the selection and selection expression layer) were involved in an exhaustive search. Another potential advantage is in the use of multiple agents trained on different feature sets, which enables a fusing algorithm to be used to enhance classification.

We now introduce the architecture in terms of semiotics, a pragmatic philosophy of signs and their role in hermeneutic acts, that is, acts of interpretation.

Send correspondence to : E-mail: dwj95r@ecs.soton.ac.uk

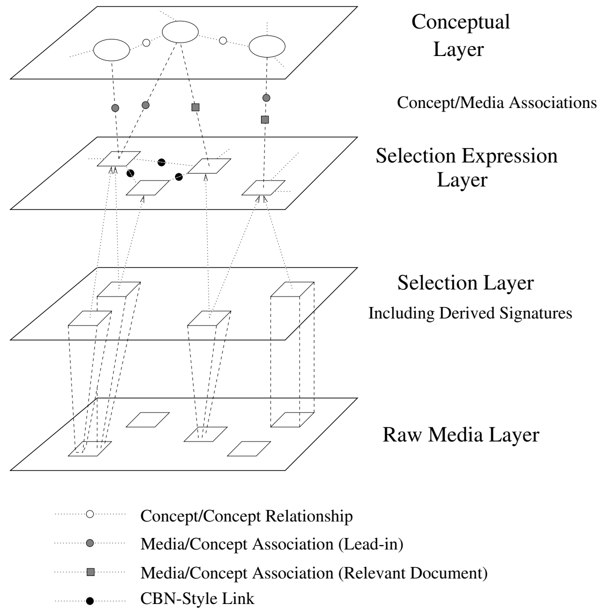


Figure 1. MAVIS2's Four Layer Multimedia Data Architecture

3. THE SEMIOTIC PERSPECTIVE

The application of semiotic theories to computer science has been covered elsewhere.[4] However, it is important to note that the role of different notions in a semiotic theory (notably, the concept of *signification* and *sign*) differ depending on the focus of the application. For example, what constitutes a *sign* in an analysis of user interfaces is not the same as a sign in the interpretation of bits as pixels. However a *sign* is the principal concept shared between all applications and theories of semiotics, but the characteristics vary between applications and different philosophical traditions model this idea differently. Here, we will focus on a level of interpretation which represents a mixture of database and image analysis and define the components of a sign appropriately.

3.1. Two Concepts of "Sign"

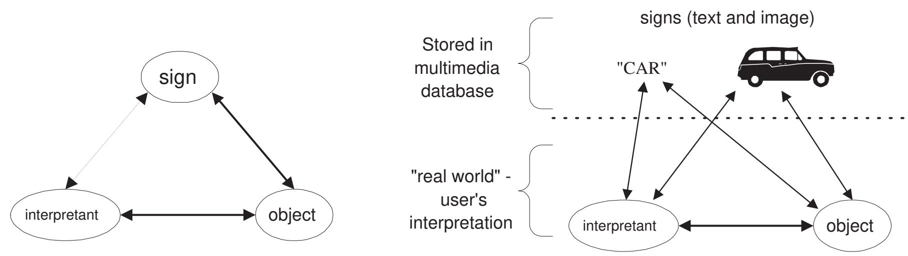
Smoliar *et al* proposed that the Saussurian notion of *signifier* and *signified* be focal concepts in mutltimedia data. The concept is easily explained. An image or text which describes a car does so by standing in a certain correspondance with familiar (usually socially agreed) notions of *what a car is* visually and metaphorically. If an arbitrary image looks sufficiently like a car, then it *signifies* a car. Loosely, the image stands as a *signifier* for the *signified* – the car. Smoliar *et al* realised that most multimedia systems only function in the domain of the signifiers, that is, on arbitrary digitally stored representations which act as signifiers only when the user is involved in a hermeneutic act.

However, we note that the "user" is abstracted out of the Saussurian model. For this reason, the approach developed by C.S. Pierce* and similarly by Ogden and Richards[6] has more use in concrete applications[7] such as multimedia. A simple interpretation involving the user of the sign is given as follows:

- the *sign* : refers to something other than itself (e.g. the word "car" or an image of a car)
- the *object* : the referrent of the sign (e.g. an actual car or class of car-like things in the world)
- the *interpretant* : the effect or causation of the concept in some user's mind (e.g. the effect of observing the sign causing a person to mentally conceive of a "car" based on their experiences of cars)

For a moment, we turn our attention to how a sign might manifest itself in a database or any other organised data structure. Figure 2 a shows the Piercean model of signs and 2 b shows the concrete example of a relationships for the car example. For ease of exposition, we might say the "real-world" depicted is the user, but it should be noted that

*A useful first reference for Pierce's ideas is J. Sowa's book[5]



a **b**

Figure 2. (a) the Piercean triadic model of signs (b) an example of "cars"

Pierce intended no such mentalistic or invidual *possession of concept* interpretation. Instead, semiotics emphasises the holism of social and cognitive factors in the whole process of sign creation and interpretation (semiosis).

Now, the scenario depicted in Figure 2 b shows how different signs are interpreted, aggregated and classified together despite the very different manifestations in the database (image and text respectively). The fact that human cognition and social experience enables such classifications leads us to the suggestion that both high-level representations (as abstract as the word "car") and what are seen as lower-level perceptual features (those present or conveyed by the image) are used conjointly in the hermeneutic act.

3.2. In Other Disciplines

In cognitive science and AI, there is a distinction between sub-symbolic and symbolic processing. This debate is still ferociously fought by proponents of either school but pragmatically, we can see a desire for a hybrid system exploiting both facets of human perception and cognition. It is precisely this kind of application where semiotics provides an elegant language for discussing and analysing the role of a symbolic ontology and its integration with low-level perceptual features; both a symbol in the ontology and the low-level perceptual features are instances of signs. The difficulty is in unifying the disjoint digital stored representations in the absence of an agent (the user) with whom the signified (the objects and interpretant) traditionally emerge. That is to say, stored signs (images, text etc.) have no internal semantics other than those conferred upon them when they are displayed to a user – on such an occasion, the signs then stand in correspondance with the Piercean triangle – we might say, the whole semiotic system embodied in Figure 2 a and b is instantiated only when a user is present.

4. MULTIMEDIA INTEGRATION AS SEMIOTICS

Let us consider how "manifestations" of objects appear in electronically stored multimedia databases. If we accept that any stored representation of an object is at best a signifier, then the burden of interpretation is solely with the user. In order that a multimedia retrieval and navigation system be ultimately endowed with a facsimile of the hermeneutic capability that enables the classification and recognition illustrated in Figure 2 b, we must consider what *kinds* of signifiers and signs are present in a multimedia system.

In order to illustrate this, we turn to a brief description of our system, MAVIS2. Figure 1 shows the type of data structure used to capture ontologies in MAVIS2. If everything is considered a sign, then the conceptual layer contains signs which conveniently represent the objects (the referrent). In MAVIS2, the conceptual layer consists of a preferred representation of each object captured by the ontology. We use textual (symbolic) signs in the conceptual layer, primarily for convenience in authoring and displaying the conceptual layer.

At lower levels, we move to the selection layer. This contains signature (feature extracted representations) of media-level objects such as images, video and audio. The vertical link between the conceptual and selection layers is mediated by a selection expression layer, which represents the connection between what might loosely be called the "concept" and its representation in a variety of media types (image, audio etc.).

Many types of signs exist in the Piercean framework, but some are less relevant here because they cope with either explicit sensory relationships (the so called *material* trichotomy) or super-symbolic notions such as law-like relationships governing signs as evidence of existence (the so called *formal* trichotomy).

The signs present in the conceptual layer are textual, symbolic descriptions – classically called high-level knowledge representations such as the word "car". In Pierce's semiotic framework, these are called *symbols*. We emphasise this term because it has technical meaning: a symbol *signifies* its object *relationally* – that is, by some agreed law or association. Other possibilities in this classification of signs include signification by *icon*, where the relation is by virtue of the sign's apparent perceptual similarity to the referrent. However, as far as textual labels for referrents in a multimedia system, they most closely resemble symbols. Symbols were (in Pierce's terminology) signs by virtue of habituation – that is, the association is learned through continued and persistent use. They differ from *icon* and *index* signs because they're visual appearance (material and indexical properties) do not affect the interpretation of the sign. It is, therefore, an arbitrary association between sign and referrant.

Associated with each symbol are the multimedia representations of the same object. For example, the word car is commonly associated with the image of a car. Such images are iconic symbols. However, we need to consider another level of signification, because the machine will assess retrieval based not on the icon, but *feature* signs extracted from the icon. This makes a semiotic model more complex, but not intangible.

Let I_c be an image or multimedia object which is a member of an aggregated collection of multimedia objects to form a conjoint representation of some real world object, captured in the conceptual layer (ontology) [†].

Let S be some function which yields a *signature* or feature-based representation of the image. Then $S_c = S(I_c)$ denotes the formal relationship between the original sign (the image of the car) and a new, feature sign which will be used in processing this image (e.g. to assess its similarity in a search). Any traditional matching algorithm will be utilising the feature signs, not the original sign. Two principles emerge:

- the *value* of the feature sign in referring unequivocally to the original sign (the image of the car) is given by its ability to represent features which enable the matching algorithm to qualitatively agree with human judgements of perceptual similarity. For example, a "poor" feature set or extraction algorithm is one which represents many perceptually different signs as the same or a very similar feature sign.
- the matching algorithms may not have access to the original sign, only the feature sign. Therefore, it is the responsibility of the feature sign to represent information (for the matching algorithm) relevant to the process of signification of the original sign.

If we hypothesise a function $\Gamma(a, b)$ which measures the similarity of two signs, and then we are presented with a query sign (e.g. another image of a different car) Q_c, then we might say that: a human assesses similarity of two images by "computing" $\Gamma(I_c, Q_c)$ whereas the algorithm implemented in a multimedia retrieval system computes $\Gamma\left(S(I_c), S(Q_c)\right)$. Hence, the machine is using a feature sign not the original sign.

Figure 3 shows how this is modelled in a semiotic framework. The signature generator S produces a vector such as $\langle s_1, s_2, \ldots, s_p\rangle$ where each of the p elements is a measure of a particular feature of the original sign. For this reason, we may treat them as "raw" indicators of specific content in the original sign (the image) such as "frequency of horizontal edges". This means that as signs, the features each represent elements of a *qualisign* – a material quality of the object it refers to. The reader is reminded that the object it refers to is in fact the sign or image of the car, not the car itself. Then, the interpretant has no intuitive meaning, but for symmetry, becomes the sum of the features and algorithms which relate the object (the image of the car) to the sign (the signature).

5. RELATED WORK

In terms of retrieving multimedia information, Santini and Jain[8] recognised that the semantics of an image are relative to the retrieval task and the user's goals, however, they did not connect this observation with semiotic notions. The novelty in Santini and Jain's work was in the way the user constructed a display space which visuo-spatially reflected the user's assessment of image similarity and relevance to the search task. In the semiotic model advocated here, the user is explicitly reflecting to the search engine their assessment of the role of the object and interpretant (signified) in relation to other signs. This is a radical advance acknowledging the role of hermeneutic acts in the user's retrieval

[†]We use images as an example, but the semiotic framework presented here applies across all media. It should be noted that the nature of the original sign might be different – a sound might reasonably be a qualisign; a purely material property pertaining to the car – but the processing performed by the machine will not be conducted on this qualisign either, but a feature sign derived from it.

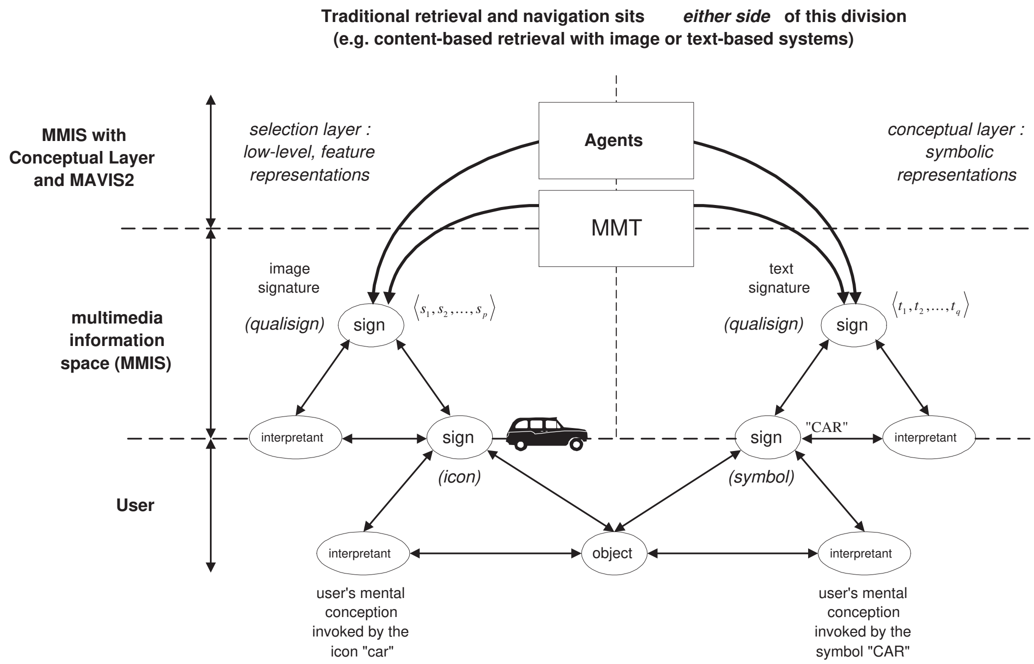


Figure 3. A Semiotic Framework for Multimedia Retrieval and Navigation

task, since most other systems ostensibly retrieve based on assumed objective geometric measures of sign similarity and a simple correspondance to object and interpretant. Other systems have attempted the kind of integration of multimedia data presented in MAVIS2.[9] However, only Smoliar *et al*[2] have connected multimedia generally with research in computational semiotics.

6. BRIDGING THE GAP

In MAVIS2, we have collections of media objects which are taken to be in some way representative of a set of concepts. Bound to these collections is an ontology describing relationships between the concepts. The conceptual layer represents domain specific information about concepts – the signs are textual descriptors. The lower selection layer consists of qualisigns of the corresponding media objects such as images of cars. In Figure 3, this division is shown as a vertical divide in the top layer of the semiotic hierarchy. We wish to bridge this gap effectively, allowing for:

- media to symbolic retrieval – the system is presented with a sign, e.g. the query image, which is converted to a set of qualisigns. Retrieval tasks take place over the set of all available qualisigns (e.g. signatures of images) and the the system returns (using the associations stored in the conceptual layer) a set of symbols indicating the conceptual association of the query image. The user then has a route into the ontology described by the conceptual layer purely by symbols (text). A user may then navigate in the conceptual layer.
- cross-media retrieval – once a query has been classified, alternative media representations (signs at the top of the bottom layer of Figure 3) can be presented to the user.

In relation to the MAVIS2 data-structure and the role of the agents, Figure 3 shows the complete semiotic system of signs in MAVIS2. The reader is reminded that the signatures are effectively signs, where the object (referrant) is the original image. The exception is text; where we have shown the text signature, in fact, it is the *preferred representation* for the MAVIS2 conceptual layer. The signature is, for example, the stored numerical representation of the word (sign) "CAR".

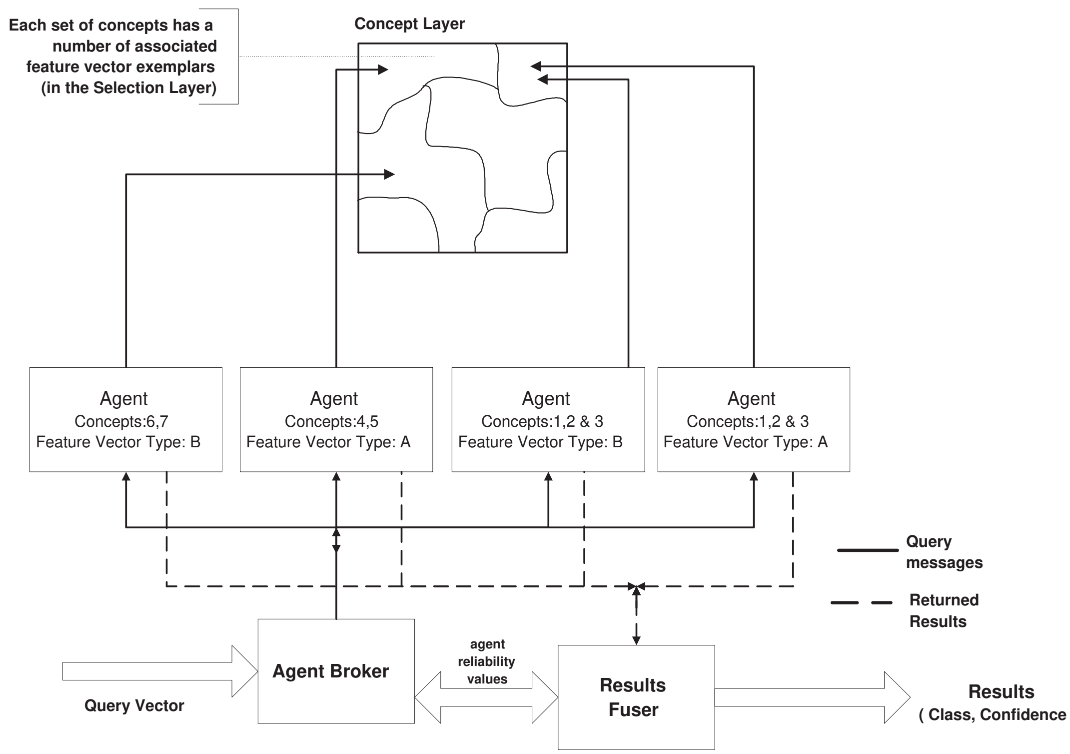


Figure 4. Multi-agent System for Media-to-Concept Classification

7. A MULTI-AGENT SYSTEM

As shown in Figure 4, each agent possesses knowledge of (and specialises on) :

- specific concept classes (signs that are symbols, signifying the object such as "car")
- specific signature types (e.g. texture-based features, spatial colour histograms)

Hence, during incremental authoring of a concept-based multimedia application, new concepts and any signatures associated are presented to an applicable, capable agent or a new agent is spawned. Dividing the agent's responsibilities by concept is shown in Figure 4.

During the query process, agents receive messages requesting classifications. These are signatures and their associated type. The agents return confidences and classifications which are then fused (using a simple weighted sum) and the MAVIS2 system then presents the agent's suggestions to the user. The user can then (if desired) provide feedback, which reflects their satisfaction with each declaration provided by an agent. This information is used to assign credit or blame to individual members of the agent population. In future, during results fusion, the trust associated with each agent is used to weight its declarations. Selectionist mechanisms such as these have been used in other multi-agent systems, and agent-based multimedia environments.[10]

An agent should pro-actively attempt to fulfill its design goals.[11] Here, agents should attempt to answer with suggested symbols (classifications and conceptual associations) when prompted by a query from the remainder of the MAVIS2 system. This may be the result of a user submitting a query.

8. CLASSIFIERS AND CONNECTIONIST TECHNIQUES

As described above, each agent is a single unit coping with a particular region of the concept-space. The agents "know" about members of the concept-space and the preferred sign (text descriptions) and has learned associations with qualisigns at the computational level of signatures. This represents the application of inductive learning to concept acquisition[12] in an ill-defined problem.

Neural network-based classification systems usually offer sub-optimal but time efficient performance (for example, when compared with optimal solutions such as first nearest neighbour) and partial freedom from explicit statistical models. This makes them candidates for a multimedia content based retrieval or navigation system, where a compromise between optimality and speed favours time performance for user satisfaction. Our classifier divides the problem in two: a system which effectively models the probability density function (pdf) of the M dimensional feature space in $N << M$ dimensions, and a module which learns to predict class/concept associations by mapping from the N dimensional space to classifications. Before describing such a system, we note that in knowledge representation terms, the neural network maps qualisigns in the computational domain to symbol signs (specifically, textual symbols) in the knowledge domain.

9. REQUIREMENTS AND CONSTRAINTS

The theory of connectionist techniques for classification is well documented.[13] However, we note the following requirements of a neural network-based classification techniques which is required to function in an agent:

1. be *constructive* : the concept-layer and selection-layer may change as a result of incremental authoring. Hence, a classifier must be able to learn new exemplar / concept associations requiring the network to grow.
2. be capable of *on-line training*: the network must be able to switch between training and performance phases.
3. be robust to noise, but sensitive to novel data: the ability to classify outlier feature vectors as either new classes or extensions to existing classes.
4. enable a control *superstructure* to be integrated : the network must be modular enough to be controlled during the training process so that correct components are trained correctly (e.g. the pdf model or the prediction generator).
5. be as far as possible, model-free : the implicit statistical assumptions should be minimal, so the classifier network is able to model heterogeneous distributions and classification schemata.

The classic MLP provides robust, analytically tractable classification. However, to obtain robust performance, training must be off-line and iterative – minimising some error measure which also dictates the credit-assignment criteria. The result for any given error measure and training regime is a classifier with deeply embedded statistical assumptions. For example, classic MLPs[14] minimise the mean-squared error so the network approximates the conditional mean of the target space.[13] If the target classifications are not normally distributed (and they rarely are in classification tasks) the predictive capability of the MLP is biased by an incorrect model of the underlying distributions. This can lead to problems if the network is asked to generalise and respond to unseen vectors (that is, to predict based on prior experience with exemplars). Also, the parameters of the MLP are difficult to optimise on-line and autonomously, the number of hidden units and learning rate being the most notable examples.

The training of any internal classification mechanism for an agent would require that it cope with statistically sparse data – multimedia applications may not generate enough batch data to train a reliable classifier. If this is the case, an agent should be able to perform *some* classification and be prepared to refine its knowledge when more data becomes available. For example, a general rule[15] is that to obtain robust classification, there should be three times as many examples (signatures) per class as there are features. Any autonomous agent responsible for the tasks defined above will need an approach grounded in principles of learning which the classic MLP does not accommodate. For this reason, we devised a general neural network and agent architecture to cope with such problem domains[16] and specialised it for the classification task presented here.

9.1. A Gaussian ART Module: Mapping M-dimensional Data into a N-dimensional Space

Figure 5 shows the basic agent architecture used in the agents presented here. The categorisation module relies on the adaptive resonance technique (ART) of Carpenter and Grossberg.[17] The extensions to the theory which make it suitable for machine learning and statistical classification tasks are presented as the Gaussian ART (GART)[18] module, and we then augmented this with our own prediction module. The GART module self organises and grows units as and when the probability density function of the input space demands, that is, when the feature space changes or is added to, the network adapts to model it as closely as the Gaussian kernels allow.

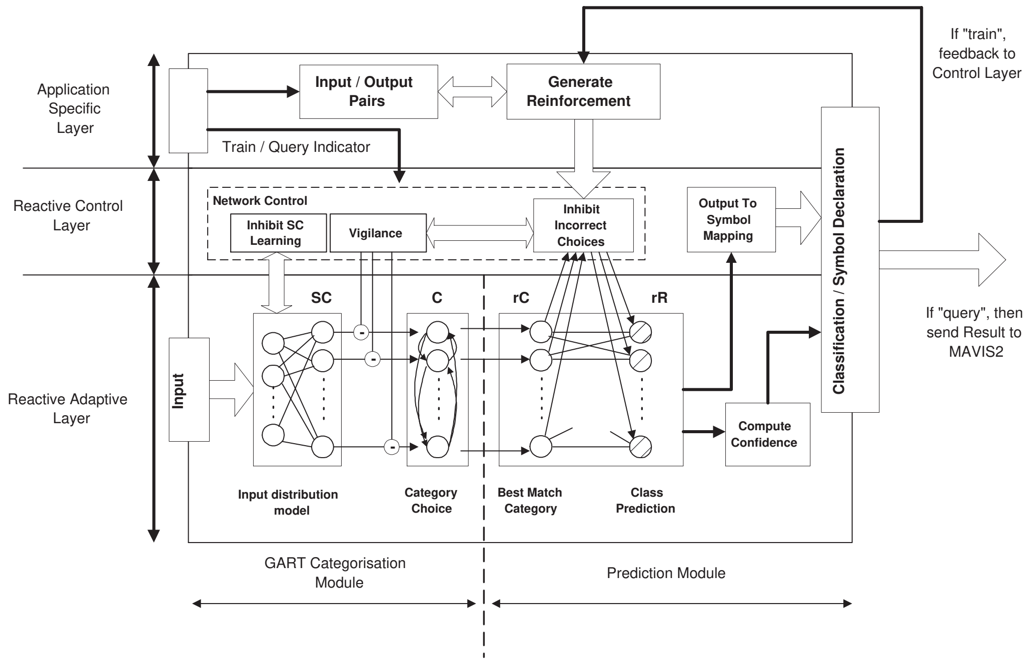


Figure 5. Agent Architecture Incorporating Neural Network for Classification Tasks in MAVIS2

At any moment in time, the agent can be switched between training and performance modes. The algorithm for processing an input is identical in both modes. The categorisation module uses a *vigilance* parameter $v \in [0, 1]$ to determine how precise it should be when matching an input to an existing category node. At $v = 1$, the agent will only recognise or classify an input if it matches precisely an existing category node in the Category Choice layer. As $v \to 0$ the agent relaxes this constraint and will match against categories which are not as close to a category node. Naturally, this has the potential to be less accurate. To perform the matching autonomously, the agent architecture has a vigilance control module which always attempts to find the best matching category at the highest vigilance. The vigilance at which the match was obtained is then used as an indicator of the agent's confidence in the categorisation. If no matching node is found after searching over the range of v, the agent returns to a baseline vigilance and then creates a new node to cope with the novel input. This forms a new Gaussian hyper-ellipsoid region in the feature space (e.g. a category node) with mean equal to the novel input and variance arbitrarily defined [‡].

9.2. Classification Prediction Module: Mapping N-categories to an Arbitrary Number of Symbols

During training, the agent performs categorisation, and the resulting winning category node activates one of the rC nodes in the prediction module. The activations are then spread in two directions: horizontally via the fully connected weights to the rR nodes, and vertically to the Control Layer which stores an "internal state" indicating the response nodes which should be prevented from activating in response to this category choice. This is because the neural network employed here was devised as a more generic model suitable for adaptive behaviour in agents. We have therefore chosen to use reinforcement learning principles to search for the correct response (given an indication of the consequence of the previous response, but no indication of the *desired* correct response) instead of the traditional error calculations of the back-propagation learning algorithm (where learning is a function of the actual and desired response). The neural network employs local representations and the weight learning is inspired by reinforcement learning principles. In combination, this enables us to grow the network as new data and class information is made available to the agent. The negative repercussions of this design choice is that we must provide training data as a data structure simulating an environment which gives positive reinforcement only when the agent responds with the correct choice.

[‡] In our experiments, the variance is set to 1 enabling us to initially ignore this parameter when configuring agents for particular classification tasks

9.3. Feedforward Prediction

The prediction module can be defined formally as follows. The winning category node excites the corresponding node rC_i in the prediction module. This spreads activation to the control layer which we will represent as a "drive node" which evaluates reinforcement provided by the training data to indicate a successful or unsuccessful prediction. We denote the activation of this node as a_D.

The nodes in the rR layer indicate the response chosen. These are then passed to a look-up table which is purely an algorithmic device to report the symbol associated with the node (e.g. the text describing the class). The choice of an rR node utilises the notion of feedforward lateral inhibition for contrast enhancement in the response nodes. If a node rR_j is activated strongly by a category rC_i then rR_j forces the activations of all $rR_{k\neq j}$ nodes down, preventing them from becoming significantly activated. This can be algorithmically implemented by softmax-type activation functions. Let the signal (output) emanating from the winning rC node be $S_i = 1$ and I_j and E_j be, respectively, the inhibitory and excitatory input to node rR_j. The activation of the response nodes rR_j is then as follows:

$$
\begin{aligned}
I_j &= S_D w_{Dj} && (1)\\
E_j &= \sum_i (w_{ij} S_i) && (2)\\
a_{rR_j} &= \frac{E_j - I_j}{L_D \sum_{k\neq j} E_k - I_k} && (3)
\end{aligned}
$$

where S_D is the activation of the drive node "inhibit incorrect choices". The activation of the drive node is determined by a *polyvalent* activation rule. This means both a reinforcement and learning signal must be present for the node to activate. For reinforcement, we denote $r = 1$ or $r = 0$ for incorrect and correct response respectively, and the train/query signal is defined as $L_D = 1$ implies training and $L_D = 0$ implies query. These signals are provided by the Control and Application specific layers of the agent architecture. Only if $r = 1$ and $L_D = 1$ does the drive node inhibit previous choices of response nodes rR. Note that during training, the lateral inhibition between rR nodes is switched on, forcing the network to choose a response node.

9.4. Inhibiting Incorrect Choices

Denoting weights from every rC node to the drive node as w_{iD}, we can specify the activation and signalling of the drive node in response to the category node as:

$$a_D = rL_D \sum_i (w_{iD} S_i) \tag{4}$$

$$S_D = \begin{cases} 1 & \text{if } a_D > 0 \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

Effectively, the drive node only affects the response nodes if the response choice was incorrect. If it is incorrect, and the agent is in training (e.g. $L_D = 1$) it enters into a cycle of searching for the correct answer (e.g. until the reinforcement $r = 0$). This is effected by quick learning of the weights w_{iD} and w_{Dj} which are then reset when a correct response is chosen. The full dynamics of such a mechanism to implement this searching process are differential equations over the weights from the category nodes to the drive, and from the drive to the response nodes. Such a mechanism enables gradual association of reinforcement with correct responses over a number of trials in response to inputs. For this application, we accelerate this learning so that this gradual association is supplanted by instantaneous association of reinforcement with response nodes and the category nodes. A response node rR_j will be selected as a result of equations 2 through 5 and category node rC_i being activated at some cycle t. If this yields negative reinforcement, then at cycle $t + 1$ the weights *from* the category node w_{iD} to the drive and from the drive *to* the response node rR_j must inhibit rR_j being selected activated again. This is effected as follows:

1. Forward propagate choice rC_i to drive node (via w_{iD}) and to response nodes rR (via weights w_{ij})
2. Calculate activations of response nodes: equations 2 through 6
3. "Winner takes all" selection of most likely response:

(a) Select highest signal S_{rR_j} and call j the winning node

(b) Set signals on all other nodes $k \neq j$ $S_{rR_k} = 0$

4. Assess reinforcement consequence (e.g. if j is the correct classification, then $r = 0$ else $r = 1$)

5. Update weights to drive node: $w_{iD} = S_D S_{rC_i} = 1$ so that the weight from chosen category node i equals unity

6. Update weights associating rC and rR nodes, w_{ij}: $w_{ij}(t+1) = w_{ij}(t) + \eta S_{rR_j} S_{rC_i}$, where $\eta = -1$ iff $r = 1$ and $\eta = 1$ otherwise.

7. If reinforcement is negative, goto 1.

The net result of this algorithm is that while learning, the response network suppresses w_{ij} where the choice of j is incorrect (e.g. generates negative reinforcement) and increases w_{ij} when correct. Step 7 above causes the whole system to cycle until the reinforcement is positive. Step 5 guarantees that the last response selection will be inhibited from being selected again during this search for the correct response.

The reader is reminded that this is a specialisation of a generalised architecture for autonomous software agents. Hence, the reinforcement-based algorithm above enables the network to be grown (because of the localisation of weight/node dependencies). The environment (providing correct classifications and reinforcements) could be replaced with a simpler mechanism that learns associations by providing the desired correct response node with high activation simultaneously with the GART module providing the categorisation of the input. We have noted, however, that this algorithm provides no significant time performance overheads in the domains tested.

9.5. Final Output Signals

During both training and query modes, the agent will declare an output based on the winning rR node. The final output of the prediction network is governed by each rR node using the following rule:

$$S_{rR_j} = \max\left\{0, \left(1 - \exp\left(-a_{rR_j}\right)\right)\right\} \tag{6}$$

Note that during learning (step 3 of the above algorithm) the control layer will suppress the activation of all but the winning node j. However, during query, all nodes output according to the function above, which merely limits the otherwise unbounded activation values into the range $[0, 1]$. During a query, the winning node is reported along with its deviation from the mean level of signalling of all nodes. This provides both the classification the agent has arrived at, plus some indication of its confidence in this classification; the rationale being that if the reported classification is not significantly higher than the other nodes, then its confidence in this classification is low.

9.6. Comparative Classification Performance

In this section, we compare the pattern classification performance of the agent proposed and regular neural network techniques. The agent's performance is compared with a regular multi-layer perceptron (trained using the least-mean-square error function) and a similar network, but with a cross-entropy independent attributes error measure[13] (which allows classifications to be reported as confidences based on the presence of features in the input).

The parameter space of the MLP techniques is taken to be the number of hidden nodes and the learning rate parameter. We tested combinations of 2, 4, 8, 16, 32 and 64 hidden nodes with learning rates of 0.01, 0.1, 0.5 and 0.9. In total, both least mean squares and cross-entropy networks are tested with one of 24 possible network configurations.

Our agent has a simpler parameter space, namely the vigilance parameter, which dictates how precise we would like categorisation to be. For comparative tests, we tested the parameter at ten values in the range [0,1]. Note that because the algorithm is constructive, no other parameters need be set by hand. The agent's control layer automates the process.

The data sets chosen were: a) iris data set from UCI[19] b) HSV-based histogram features from a MAVIS2 data set c) RGB-based histogram features from a MAVIS2 dataset d) 3 clusters of Gaussian distributed data. The rationale was as follows : a) is a standard machine learning data set, and is of relatively compact feature dimensionality with adequate samples. Both b) and c) are grossly under-determined datasets, each containing 208 samples (one per image) of 27 and 96 features respectively – this represents a case where the agent must try its best to perform classification in a difficult multimedia problem. Test set d) is a simple two-dimensional problem, which contains enough samples (150) to train on and is present to test whether the classifiers and agent are working as expected.

Recall from section 9, that the agent must perform even in the absence of "good" data which would enable robust classification. An autonomous agent should attempt to provide the classification service even when the data is weak. It should be capable of learning incrementally, so it can improve if or when more data becomes available. In semiotic terms, the agent must cope with sparse instances of icons and corresponding qualisigns but still enable the user to be presented with symbols.

Each result presented is the average over 10-fold cross-validation testing[20]; this enables us to assess the "memorisation" of the training set and (more significantly) its generalisation performance on unseen data. Validation sets are generated from 10 independent samplings drawn from the original data and are kept back during training. This ensures that generalisation performance on unseen queries is not attributable to fortuitous choice of a hand-picked validation set.

We present the results in the table below. The first column indicates the network used; multiple independent attributes and cross entropy error (MIA), standard least-mean-squares error (LMS) and our agent technique (AG). The second and third columns are best training and validation set performances over the whole parameter space tested. The remaining columns report the number of hidden nodes and learning rate (or vigilance in the case of our agent-based technique) which gave the best performances, plus the mean number of iterations required to train the networks and finally the standard deviation (s.d.) of this value. A low s.d. indicates that almost always, the best performance can be achieved in the mean number of iterations, whereas a high s.d. indicates a large variation in the number of iterations required. We argue that for autonomous agents, the training behaviour must be predictable and this value should be low.

ANN	Data Set	Best TS Performance	Best VS Performance	Hidden Nodes	Learn Rate	Best Performance at Iteration	Standard Deviation of Best Iteration
MIA	Gaussian	100%	100%	64	0.01	1	0
LMS	Gaussian	100%	100%	64	0.01	1	0
AG	Gaussian	98.9%	100%	n/a	0.6	1.3	0.4
MIA	iris	94.5%	99.3%	64	0.9	4.4	3.6
LMS	iris	94.7%	98.6%	32	0.01	12.7	9.8
AG	iris	95.5%	99.3%	n/a	0.7	3.2	2.2
MIA	HSV	50%	50%	64	0.1	9.5	6.5
LMS	HSV	45%	49%	32	0.01	17.7	12.5
AG	HSV	24.4%	39%	n/a	0.2	3.9	3.1
MIA	RGB	28%	46%	8	0.5	20.3	15.3
LMS	RGB	32.6%	45%	32	0.01	9.6	11.9
AG	RGB	29.9%	38%	n/a	0.7	3.2	2.2

The table shows that for all data sets except the artificial pure Gaussian data the s.d. at best iteration is lowest for our Agent Technique. However, the classification performance of the technique in its present form is not quite matching the MLP approaches and we are doing further work to tune the technique in this area.

10. CONCLUSION

The semiotic framework presented enables the treatment of multimedia information and the over-worked term "semantics" in a principled framework. It highlights opportunities for novel treatment of different media and demarcates philosophical boundaries that exist and have been typically encountered in engineering of multimedia systems. We believe that generally, it correctly formalises the relationships between nebulous concepts such as ontology and the representation of symbolic knowledge in multimedia systems. We differ from Smoliar *et al*[2] in that we prefer the triadic model, which explicitly involves the user and the system's processing in the analysis.

An agent based approach was offered as a metaphor for reasoning about individual systems which attempt to bridge the gap between high-level and low-level multimedia data. Neural network based classification was offered as one possible method for implementing the kind of sign generalisations that humans are typically capable of. We later discovered that independently, a similar proposal was given by John Sowa in his forthcoming book.[21] The limitations of our approach are demonstrated by the semiotic framework: that classification is only as sound as the feature signs used by virtue of the relationship between the signs.

Our experiments with different neural networks to implement the agent's have revealed that the GART-based technique offers the requirements of section 9, and has significantly more predictable training behaviour and performs favourably against the MLP variants on most problems. However, on ill-defined problems (e.g. the HSV and RGB data), more work is required to enable it to compete with the classic techniques.

Future work will consider novel ways of manipulating the relational icons, indexes and symbols which provide more information to the agents, enabling them to more usefully contribute to the searching process. In terms of agents and classification, additional functionality must be added to the application specific layer to enable better performance on sparse data sets.

11. ACKNOWLEDGEMENTS

The authors wish to thank the EPSRC for the support of Grant Number GR/L03446.

REFERENCES

1. M. Dobie, R. Tansley, D. Joyce, M. Weal, P. Lewis, and W. Hall, "A flexible architecture for content and concept based multimedia information exploration," in *Proceedings of the Challenge of Information Retrieval*, 1999. Also available from `http://www.mmrg.ecs.soton.ac.uk/publications/archive/dobie1999/`.
2. S. W. Smoliar, J. D. Baker, T. Nakayama, and L. Wilcox, "Multimedia search: An authoring perspective," in *Proceedings of Image Databases and Multi-Media Search*, pp. 1–8, 1996.
3. S. Harnad, "The symbol grounding problem," *Physica D* **42**, pp. 335–346, 1990.
4. P. B. Andersen, "Computer semiotics," *Scandinavian Journal of Information Systems* **4**, pp. 3–30, 1992.
5. J. F. Sowa, *Conceptual structures: Information Processing in Mind and Machine*, Addison-Wesley Publishing Company, 1984.
6. C. K. Ogden and I. A. Richards, *The Meaning of Meaning*, Routledge and Keagan Paul, London, 2 ed., 1949.
7. P. B. Andersen, *A Theory of Computer Semiotics*, Cambridge University Press, 1997.
8. S. Santini and R. Jain, "Interfaces for emergent semantics in multimedia databases," in *Proceedings of the IS&T / SPIE Conference on Storage and Retrieval for Image and Video Databases*, pp. 167–175, 1999.
9. K. Hirata, S. Mukherjea, Y. Okamura, W.-S. Li, and Y. Hara, "Object-based navigation : An intuitive navigation style for content-oriented integration environment," in *Proceedings of ACM Hypertext 1997*, pp. 75–86, 1997.
10. T. P. Minka and R. W. Picard, "Interactive learning with a society of models," *Pattern Recognition* **30**(4), pp. 565–581, 1997.
11. N. R. Jennings, K. Sycara, and M. Wooldridge, "A roadmap of agent research and development," *Autonomous Agents and Multi-Agent Systems* **1**, pp. 275–306, 1998.
12. R. S. Michalski, J. G. Carbonell, and T. M. Mitchell, *Machine Learning – An artificial intelligence approach*, Tioga Publishing Company, Palo Alto, California, 1983.
13. C. M. Bishop, *Neural Networks for Pattern Recognition*, Oxford University Press, 1995.
14. P. Radhakrishan and P. H. Lewis, ""Agent Like" support for content based retrieval and navigation," in *Proceedings of SPIE Storage and retrieval for image and video databases VI*, pp. 228–235, Jan. 1998.
15. D. H. Foley, "Considerations of sample and feature size," *IEEE Transactions on Information Theory* **18**, pp. 618–626, sept 1972.
16. D. W. Joyce and P. H. Lewis, "Connectionist models for learning choice behaviour in reactive software agents," in *Proc. of the IASTED International Conf. on Artificial Intelligence and Soft Computing*, pp. 437–442, 1999.
17. G. A. Carpenter and S. Grossberg, "A massively parallel architecture for a self organising neural pattern recognition machine," *Computer Vision, Graphics and Image Processing* **37**, pp. 54–115, 1987.
18. J. R. Williamson, "Gaussian ARTMAP: A neural network for fast incremental learning of noisy multidimensional maps," *Neural Networks* **9**(5), pp. 881–897, 1996.
19. C. Blake and C. Merz, "UCI repository of machine learning databases," 1998.
20. P. R. Cohen, *Empirical Methods for Artificial Intelligence*, MIT Press, 1995.
21. J. F. Sowa, *Knowledge Representation: Logical, Philosophical and Computational Foundations*, Brooks/Cole Publishers, CA, 2000.