Hybrid Neural Document Clustering Using Guided Self-Organization and WordNet

Chihli Hung, Stefan Wermter, Peter Smith

University of Sunderland
School of Computing and Technology
Center for Hybrid Intelligent Systems
Sunderland SR6 0DD, UK
chihli.hung@sunderland.ac.uk
http://www.his.sunderland.ac.uk

Hybrid Neural Document Clustering Using Guided Self-Organization and WordNet

Chihli Hung, *University of Sunderland and De Lin Institute of Technology*

Stefan Wermter and Peter Smith, *University of Sunderland*

A guided approach to document clustering that integrates linguistic top-down knowledge from WordNet into text vector representations based on the extended significance vector weighting technique improves both classification accuracy and average quantization error.

Document clustering is text processing that groups documents with similar concepts. It's usually considered an unsupervised learning approach because there's no teacher to guide the training process, and topical information is often assumed to be unavailable. In contrast, document classification is usually considered a supervised learning approach because preclassified information guides the training process. If, however, the corpus offers topical information, both classification and clustering techniques can take advantage of words' relationships to different topical concepts with different weights. In this case, a *guided* neural network based on topical information lets users exploit the domain knowledge and reduces the gap between human topical concepts and data-driven clustering decision.

The self-organizing map is a network for guided or unguided clustering. SOM combines nonlinear projection, vector quantization (VQ), and data-clustering functions.[1] As Teuvo Kohonen and colleagues point out, "one should provide the different words with such weights that reflect their significance or power of discrimination between the topics."[2] They suggest using the vector space model (VSM) to transform documents to vectors if no topical information is provided. However, they also state: "If, however, the documents have some topical classification which contains relevant information, the words can also be weighted according to their Shannon entropy over the set of document classes."[2] In fact, their WebSOM project uses a modified VSM that includes topical information.[2,3]

Our guided self-organization approach is motivated in a similar manner but we further integrate topical and semantic information from WordNet. Because a document-training set with preclassified information implies relationships between a word and its preference class, we propose a novel document vector representation approach to extract these relationships for document clustering. Furthermore, merging statistical methods, competitive neural models, and semantic relationships from symbolic WordNet, our hybrid learning approach is robust and scales up to a real-world task of clustering 100,000 news documents.

Framework and data sets

Figure 1 shows our hybrid neural document clustering framework. The model consists of two phases. First (Figure 1a), if topical information is available, we represent documents as normalized extended significance vectors, or ESVs (our guided SOM-like model); if it isn't available, we represent documents as vector space representations (the unguided model).

Second (Figure 1b), we convert the significance vector lexicon to its n-level hypernym version by extracting knowledge from WordNet. In this step, our lexicon contains two parts: a word-hypernym look-up table and a word-topic weight matrix. We then transform the data set into significance vectors according to the n-level hypernym lexicon and build a guided SOM-like model with WordNet knowledge.

1094-7167/04/$20.00 © 2004 IEEE
Published by the IEEE Computer Society

We based our experiments on the new version of the Reuters corpus, RCV1 (see http://about.reuters.com/researchandstandards/corpus), because it's a representative test for text classification, a common benchmark, and a fairly recent comprehensive data source. The corpus contains 984 Mbytes of news articles in compressed format published between 20 August 1996 and 19 August 1997. The 806,791 news articles include 9,822,391 paragraphs, 11,522,874 sentences, and about 200 million words. The corpus defines 126 topics, but 23 of them contain no articles. It preclassifies news articles to an average of 3.17 topics each.

We initially concentrated on the eight most dominant topics for our data sets, as listed in Table 1. Because a news article can be preclassified to more than one topic, we consider the *multitopic* as a new combination of topics. Thus we expand the eight chosen topics to 40 combined topics for the first 10,000 news articles and confine any combination of topics to these 40 multitopics for both training and test sets. The first 10,000 full-text news articles are our training set and the next 10,000 full-text news articles are our test set. Three multitopics in the training set don't appear in the test set because we chose the two sets based on time (that is, given the eight most dominant topics, the first 10,000 documents are for the training set and the second 10,000 documents are for the test set).

Next, we scale up our experiments to 100,000 full-text news articles and expand the eight chosen topics to 54 multitopics. Table 2 shows the data set distribution.

Evaluation criteria

Evaluation of SOM-like modes needs careful analysis. Although we can see clusters in the SOM-like maps, we can't judge whether one SOM-like map is better than another simply by their appearance. We use SOM-like models to get an objective evaluation from the corpus itself. In this article, we evaluate models using classification accuracy (CA) and average quantization error (AQE), as have several other researchers.[2,4]

Classification accuracy

We use Kohonen and colleagues' measure of classification error[2]: "all documents that represented a minority newsgroup at any grid point were counted as classification errors … the node and the abstracts belonging to the other subsections were considered misclassi-

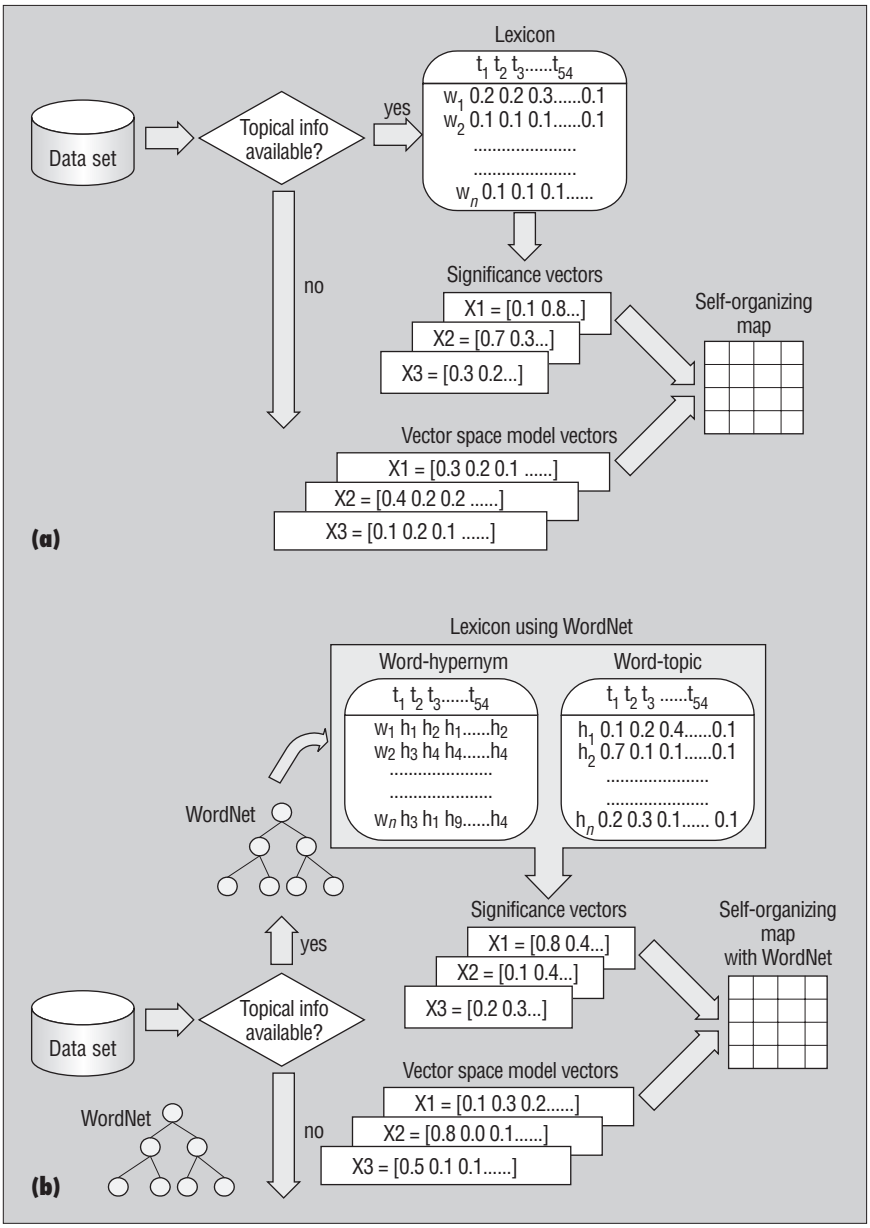

Figure 1. Hybrid neural document clustering framework: (a) in phase 1, we represent documents as guided or unguided SOM-like models; (b) in phase 2, we build a guided SOM-like model with WordNet.

Table 1. Selected topics and their distribution in the Reuters-RCV1 corpus.

Topic	Description	Distribution
c15	Performance	149,359
c151	Accounts/earnings	81,201
c152	Comment/forecasts	72,910
ccat	Corporate/industrial	372,099
ecat	Economics	116,207
gcat	Government/social	232,032
m14	Commodity markets	84,085
mcat	Markets	197,813

Table 2. Topic distribution for data sets of 10,000 and 100,000 full-text news articles.

Number	Topic composition	10,000		100,000
		Training set	Test set	Training set
1	ecat/mcat	155	104	1,034
2	ccat	1,780	2,033	20,660
3	c15/c151/ccat/ecat/gcat	6	2	32
4	c15/c151/ccat	999	916	6,530
5	m14/mcat	877	846	8,197
6	ecat	771	672	7,368
7	ccat/gcat	293	392	3,557
8	ccat/ecat/gcat	162	174	1,842
9	mcat	1,135	1,182	11,202
10	gcat	2,152	2,021	22,337
…				
39	c15/c151/ccat/gcat	1	3	37
40	c15/c152/ccat/ecat/mcat	1	0	3
…				
53	c15/c151/ccat/gcat/m14/mcat			1
54	c15/c151/c152/ccat/ecat			3

fications." After the training process, we assign a map unit label according to the highest number of assigned labels. Therefore, every unit represents its major news article labels. We replace the assigned label of each news article mapped into this unit with the unit label. Thus, the mapping is correct if the replaced input vector label matches its original label based on Reuters' multitopic classification.

We calculate CA as the proportion of correct mappings to input news articles. For example, assume 10 news articles are in the data set with one unit in a trained SOM. Three news articles have a $topic_1$ label and seven have a $topic_2$ label. All news articles map to $unit_1$ because only one unit is in this example SOM. Thus, we assign all news articles mapped to $unit_1$ the unit label $topic_2$. In this example, CA is 70 percent.

Average quantization error

AQE is a measurement used in vector quantization.[5] Also called the *distortion measure*, AQE is defined as the average distance between every input vector and its best matching unit, as described by the equation

$$AQE = \frac{1}{N}\sum_{i=1}^{N}\left\|\vec{x}_i - \vec{w}_i\right\|, \qquad (1)$$

where N is the total number of input patterns, $\vec{x}_i$ is the vector of each pattern, and $\vec{w}_i$ is the BMU vector for the input pattern i.

Small AQE values mean small distortion for all input vectors to their cluster centers, and thus the AQE is a direct index linked to the model's explanatory ability. A good clustering approach has a small AQE. We use it to judge whether the hypernym relationship contributes to the model's performance.

Vector representations

Term weighting is a well-known representation approach that transforms a term to a weight vector in automatic text clustering. For unsupervised neural models, this representation plays a key role in model performance. VSM, the most common term-weighting method, is based on the *bag-of-words* approach, which ignores the linear ordering of words within the context and uses basic occurrence information.[6]

However, because VSM's dimensionality is based on the total number of specific terms in the data set, it doesn't efficiently handle large data sets. Our full-text experiments included 7,223 open-class words (that is, nouns, verbs, adjectives, and adverbs) from 1,000 full-text news articles. For 10,000 and 100,000 full-text news articles, we had 15,760 and 28,687 specific terms. Dealing with huge text collections means dealing with huge dimensionality, which can affect efficiency.

Dimensionality reduction is one reason researchers examine alternative vector representations. Another reason is to extract important features and filter noise, and thus improve clustering performance. The most common way to do this is to omit the stop words and least common words, and to stem words to their base forms. Another frequently used method is to choose only the most frequent or most salient words from the master word list. For example, in addition to removing stop words, Hsinchun Chen and colleagues use the 1,000 most frequently used words in the text collection to build their master word list.[7] Thus, a vector with 1,000 elements represents each document.

In our work, in addition to VSMs, we apply an ESV representation approach.[8] Each word has a preference for a specific semantic category. We analyze the importance of words in each category and build a lexicon based on the relationship between words and preassigned document topics. Thus, an m-dimension vector, where m is the number of preassigned multitopics in the document, can represent a news article. This method avoids the dimensionality problem by using a domain-dependent but automatically generated lexicon that assigns each word a category behavior representation.

We remove stop words, lemmatise words to their base forms using WordNet, and restrict our experiments to words found in WordNet (which uses only open-class words). We build a semantic lexicon by collecting the word frequency for a topic and transform each word into a significance vector. We then add all significance vectors of words occurring in a document and normalize them to produce a document-ESV.

We start this process with the word-topic occurrence matrix, which we describe as

$$\begin{bmatrix} o_{11}\ o_{12}\ o_{13}\ldots o_{1M} \\ o_{21}\ o_{22}\ o_{23}\ldots\ o_{2M} \\ \ldots\ldots\ldots\ldots\ldots \\ \ldots\ldots\ldots\ldots\ldots \\ o_{N1}\ o_{N2}\ o_{N3}\ldots o_{NM} \end{bmatrix},$$

where o_{ij} is the occurrence of word i in topic j, M is the total number of topics, and N is the total number of different words. We represent an element of a significance vector for a word

i in topic j as w_{ij} and obtain it using the equation

$$w_{ij} = \frac{o_{ij}}{\sum_{\tilde{j}=1}^{M} o_{i\tilde{j}}} \quad . \tag{2}$$

The number of news documents observed for each topic influences Equation 2. We define Equation 3 as the ESV representation approach. ESV uses the logarithmic weights of the total number of word occurrences in the data set divided by the total number of word occurrences in a specific semantic topic to alleviate skewed distributions in Equation 2. Thus, we define an element in word vector $\vec{w}_i$ for topic j as

$$w_{ij} = \frac{o_{ij}}{\sum_{\tilde{j}=1}^{M} o_{i\tilde{j}}} \times \log \frac{\sum_{\tilde{i}=1}^{N}\sum_{\tilde{j}=1}^{M} o_{\tilde{i}\tilde{j}}}{\sum_{\tilde{i}=1}^{N} o_{\tilde{i}j}} \quad . \tag{3}$$

We define the news document vector $\vec{d}$ as a summation of significance word vectors $\vec{w}_i = (o_{i1} o_{i2} \ldots o_{iM})$ divided by the number of words in a document:

$$\vec{d} = \frac{1}{n} \sum \vec{w} \ , \tag{4}$$

where n is the number of words in news document d.

Extracting semantic concepts from WordNet

WordNet contains semantic relationships in *synset*, a set of synonyms representing a distinct concept.[9] Our work exploits WordNet's hypernym-hyponym relationship to determine whether we can obtain fewer but more general concepts and thus further improve SOM's classification ability.

A term's hypernym is a general term whereas a hyponym is specific. This relationship is similar to how news is categorized: a cluster of news is more general than an individual news article. We group news articles with a similar concept in the same class.

A vocabulary problem exists when a term is present in several synsets as Figure 2 shows. Determining the correct concept for an ambiguous word from several synsets is difficult, as is deciding the concept of a document containing several ambiguous terms. Darin Brezeale transforms such documents based on their binary representations.[10] He uses the first synset directly on WordNet because of the greatest frequency of occurrence in WordNet. Sam Scott and Stan Matwin use *hypernym density*—that is, the number of occurrences of the synset in the document divided by the number of words in the document—to decide which synset is more likely than others to represent the document.[11] Other researchers handle the word sense disambiguation problem manually.[12]

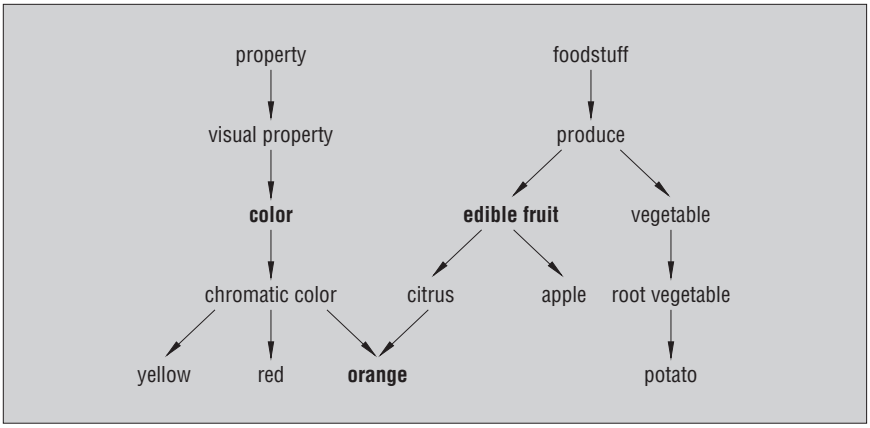

Figure 2. Two hypernym trees for the term "orange." The two-level hypernym for orange with the color concept is "color" but with the fruit concept is "edible fruit."

We do not use the synset directly; rather, we take advantage of the synset's *gloss*, which explains each concept and gives an example sentence. For example, the gloss of the word "orange" with the fruit concept is "round yellow to orange fruit of any of several citrus trees"; with the color concept it is "any of a range of colors between red and yellow."

We convert the semantic lexicon into its hypernym version word by word and topic by topic. Each ambiguous word in the original lexicon contains several senses and each sense has its own gloss. We treat each gloss as a small piece of the document with a core concept and transform the gloss using the ESV representation (Equations 3 and 4).

To determine the gloss for an ambiguous word, we compare the specific element weights of each gloss in the specific topic of the original semantic lexicon. We choose the gloss vector with the highest weights in the specific element to represent the original word. For example, we only compare the first element weight of all gloss vectors for an ambiguous word of topic 1. Then, going up n-levels in the hypernym tree, we can use this hypernym to build our hypernym version of a semantic lexicon for all terms in all topics. Thus, we replace *sibling words* (words with the same direct hypernym in WordNet) from the same topic with the same hypernym, and sibling words from different topics with different hypernyms.

The hypernym relationships in WordNet form a tree-style hierarchy. That is, more child words than parent words exist. Thus, this approach can theoretically reduce the total number of words in a data set.

We use the following example to describe our approach: The ESV of the word "orange" in the semantic lexicon is [0.234 0.033 0.502 … 0.002] and its two gloss vectors with color and fruit concepts are [0.101 0.203 0.302 … 0.031] and [0.201 0.103 0.222 … 0.021]. When we convert "orange" in topic 1, we only compare the first element for two gloss vectors. Thus, we choose the gloss with fruit concept in topic 1. When we convert "orange" in topic 2, we choose the color concept. Therefore, one word in one topic has only one hypernym tree. This differs from word sense disambiguation, which usually defines the word concept according to its context. However, we prefer using classification knowledge to improve our document-clustering model.

To use the knowledge of the semantic lexicon hypernym version, we convert each news article from its original version to its n-level hypernym version. Because we've already built a hypernym and topic look-up table (Figure 1), we can easily find a hypernym for a word in a document based on its preclassified topic. By looking up the n-level hypernym for all words in the document, we transform each document to a vector using the ESV representation based on the n-level hypernym lexicon. This approach reduces the total number of distinct words for our data sets and improves the clustering and classi-

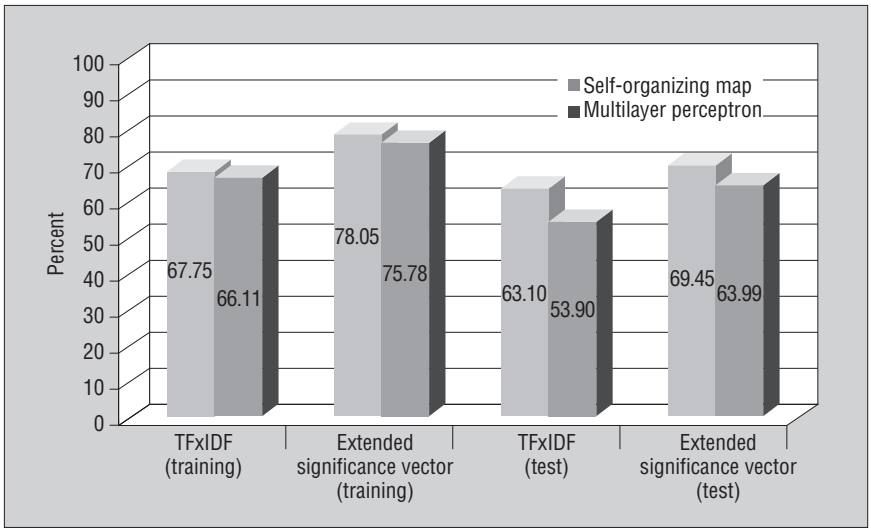

Figure 3. Comparison of two vector representation methods using classification accuracy criterion for 10,000 full-text news articles. The *y*-axis represents classification accuracy for the exact one of the 40 combined topics.

fication performance for SOM-like models. However, the test set's preclassified information shouldn't be used because the test set is the unknown data set, and its preclassified information is only for evaluation. To overcome this problem, we apply the knowledge from the original lexicon that is built based on the training set only. We then transform the test set using the significance vector representation approach (Equations 3 and 4) by looking up this lexicon. We treat the most significant element of each document vector of a test set as a topic label.

Experiments

To illustrate our hybrid model, we performed detailed comparisons with traditional approaches using our evaluation criteria. Our experiments included

- A comparison of TFxIDF (term frequency times inverse document frequency) and our ESV representation based on MLP (multilayer perceptron) and SOM using 10,000 full-text news articles
- An extension of the above comparison using the WordNet hypernym relationship
- A comparison of other SOM-like models using different hypernym levels and scaling up those models to handle 100,000 full-text news articles

TFxIDF vs. ESV with MLP and SOMs

When the vector representation isn't related to the topic information, a traditional neural model, such as MLP, can learn the relationship between the input and output units. In contrast, SOM builds its own clusters according to the input unit weights, and the clusters might differ from human labelling. If a lexicon offers topical significance vectors, however, both MLP and SOM can exploit vector representation to more clearly discriminate between topics. When we apply the corpus knowledge in the vector representation, we modify an unsupervised SOM to a SOM that uses guiding topical information for its input representation—that is, a guided SOM. We used normalized TFxIDF scheme in our experiments because it is one of the best weighting models for long document vectors.[6] TFxIDF considers words highly important if they appear often in one document but not throughout the corpus. Equation 5 describes the normalization on TFxIDF used in our representation:

$$TFxIDF\ term\ weight = \frac{w}{\sqrt{\sum_{vector} w^2}} \quad (5)$$

where w = term frequency in a document × $\log(D/d)$, D is the total number of different words in the lexicon, and d is the number of documents containing this term.

After removing stop words and lemmatizing words, we choose the 1,000 most frequent words for TFxIDF. We use a resilient back propagation training algorithm,[13] which uses a local adaptive learning scheme, for MLP because it eliminates the harmful influence of the size of partial derivatives when using a traditional sigmoid function. Therefore, resilient back propagation offers a small convergence time and greater robustness than other MLP training algorithms.

We used 8, 16, and 32 units of a hidden layer for MLP, running each architecture three times to reduce the effect of the random initial weights. A one-hidden-layer MLP with 16 hidden units gave the best results. As Figure 3 illustrates, SOM and MLP with ESV, which is based on the lexicon of significance vectors, were superior to TFxIDF in all cases. These results show clearly that the corpus topical information offers better discriminatory power for MLP and SOM. SOM exceeds MLP using ESV representation for both the training and test sets. Using this knowledge, an unsupervised guided learning algorithm can outperform a traditional supervised learning algorithm in a classification task.

TFxIDF vs. ESV with a neural model and WordNet

Next, we try to extract WordNet knowledge to make data vectors with more discriminatory power between topics. In these experiments, we use the one-level hypernym to replace the original word if its hypernym exists. We've tested other hypernym levels, but the one-level is most accurate. We used the first-sense policy[10] to choose the TFxIDF representation hypernym and our document transformation method described earlier for ESV representation. As Figure 4 shows, integrating a neural model and WordNet knowledge outperforms the isolated model.

WordNet knowledge doesn't, however, contribute much to a SOM model with TFxIDF representation based on the classification criterion. A SOM using TFxIDF is a totally unsupervised model and will produce its own clusters. Therefore, when we replace each word with its hypernym using the first-sense policy, we also convert the same word in different topics to the same hypernym, which can reduce the discriminatory power of words between topics.

The accuracy improvement for both models using ESV representation in the training set is much better than the improvement in the test set. This is because we apply the corpus topical information for the training set but not for the test set. We get the test document label from a lexicon based on the training set, which can differ from the preclassified label. The purpose of this is to apply available knowledge from the corpus

and use competitive learning approaches to analyze the domain clustering performance. As Kohonen and colleagues suggest,[2] because SOM aims to organize a given data set into a structure from which users can easily retrieve the documents, we keep the test set for consistency with the supervised method only.

Comparing models and scaling up

According to the previous experiments, using corpus knowledge improves CA for both MLP and SOM. SOM outperforms MLP on both the training set and test set when using the ESV representation. We extended our model to other competitive neural learning models as alternatives to the SOM approach. Apart from SOM, these included two other static models—competitive learning (CL) and neural gas (NG)—and three dynamic models—growing grid (GG), growing cell structure (GCS), and growing neural gas (GNG). See the "Alternate Competitive Learning Models" sidebar for descriptions of these algorithms.

We use 15 × 15 (225) units for each model, as Figure 5 illustrates. However, this unit number is only an estimate for dynamic models since dynamic models grow periodically and prune when units are unsuitable for representing input samples. According to our experiments, if we use these models alone, CA

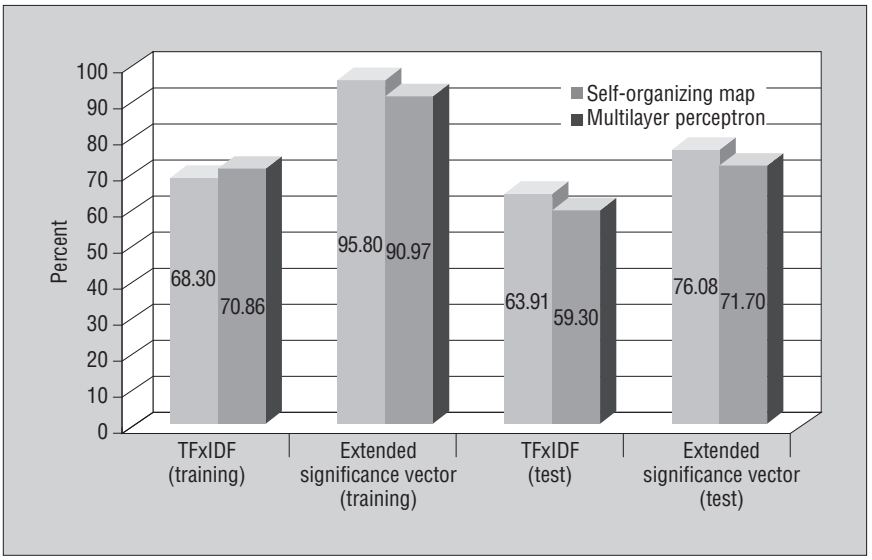

Figure 4. Comparison of two vector representation methods integrated with WordNet knowledge and based on the classification accuracy criterion for 10,000 full-text news articles.

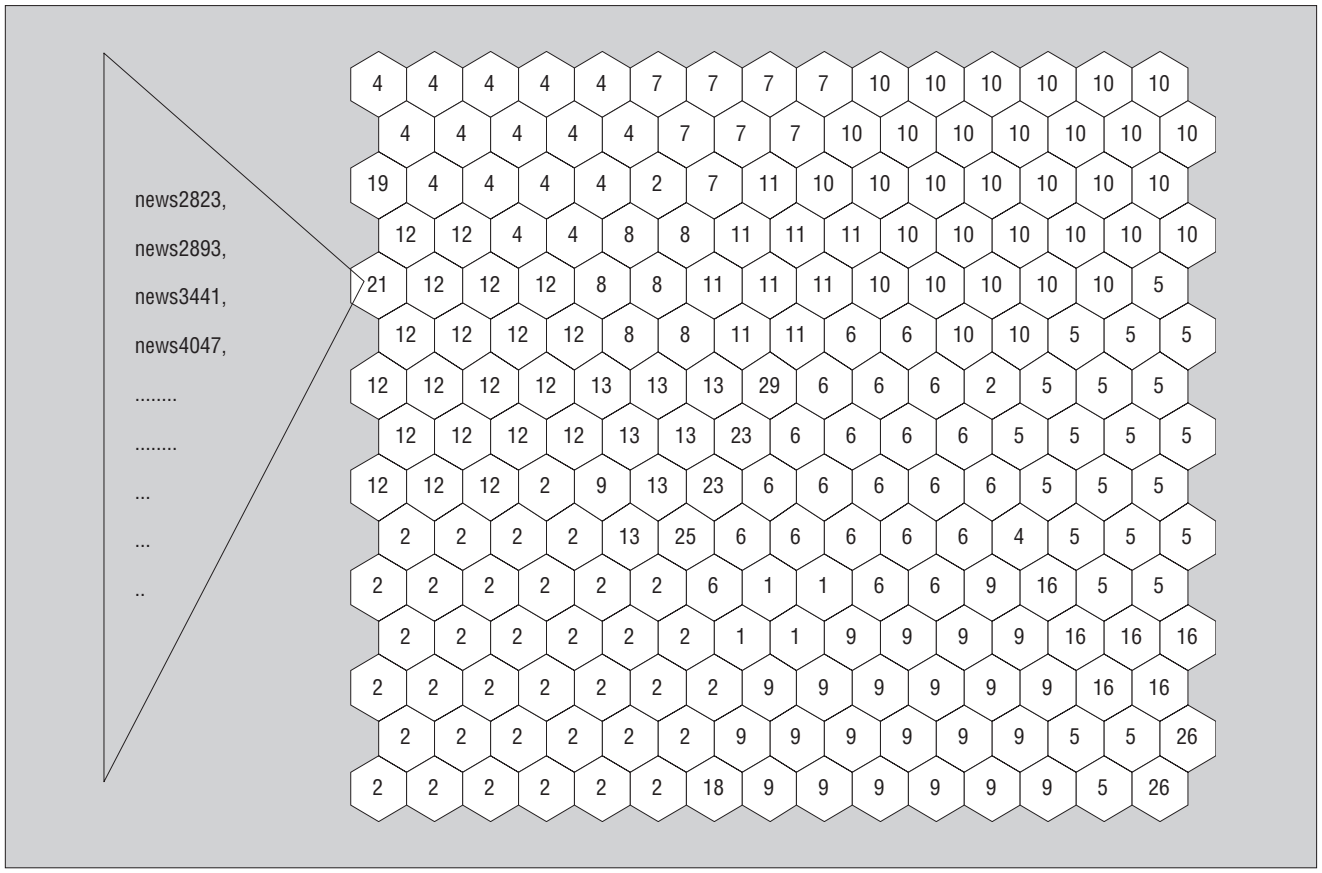

Figure 5. Static self-organizing map (SOM) model with 15 × 15 units. Reuters topic codes appear as numbers.

Alternate Competitive Learning Models

All six models we tested aim to map a data set from a high-dimensional space onto a low-dimensional space, keeping its inner structure as faithful as possible. We divide these algorithms into static and dynamic models.

Static models

We tested our approach in three static competitive learning models: competitive learning (CL),[1] self-organizing map (SOM),[2] and neural gas (NG).[3] The main difference between them is how they update their cluster centers.

- CL is the online version of k-means,[4] a traditional statistic clustering method. CL tries to keep its *k* cluster centers representing the arithmetic mean of the input vectors. Because it has no neighboring relationships among unit centers, it updates only the best matching unit (BMU)—that is, the output unit of the model with the shortest Euclidean distance to its associated input vector.
- SOM projects the high-dimensional input vectors into a 2D space. SOM uses a grid to define its neighboring boundary and relationship. It updates unit centers inside the neighboring boundary according to their distance from the input vector.
- NG is a SOM-like model with loose relationships between units, so the clusters are treated as gas, which spreads in the input space. It updates unit centers based on their distance from the input vector. Figure A shows the NG convergence using its two principal components.

Dynamic models

CL, SOM, and NG are static models because the network uses a fixed number of units. Apart from defining neighborhood differently, dynamic models have varying dynamic unit representations. This group of competitive learning algorithms automatically defines the number of units before training. According to Bernd Fritzke, a SOM model might have a good representation on the input vectors with uniform probability density, but the representation might not be ideal for complex clustering from a topology-preservation viewpoint.[5] He thus proposes a series of dynamic competitive learning models.

We tested our approach in growing grid (GG),[6] growing cell structure (GCS),[7] and growing neural gas (GNG)[8] models.

- GG is an incremental variant of a SOM in terms of model topology. It has two stages: growing and fine tuning. Its update rule is the same in both stages, but the learning rate is fixed in the growing stage and decayed in the fine-tuning stage. As Figure B shows, GG starts from 2×2 units in a grid architecture and develops the grid by columns or rows. GG generates the units from the position between the most frequently activated unit and its farthest direct neighbor unit.

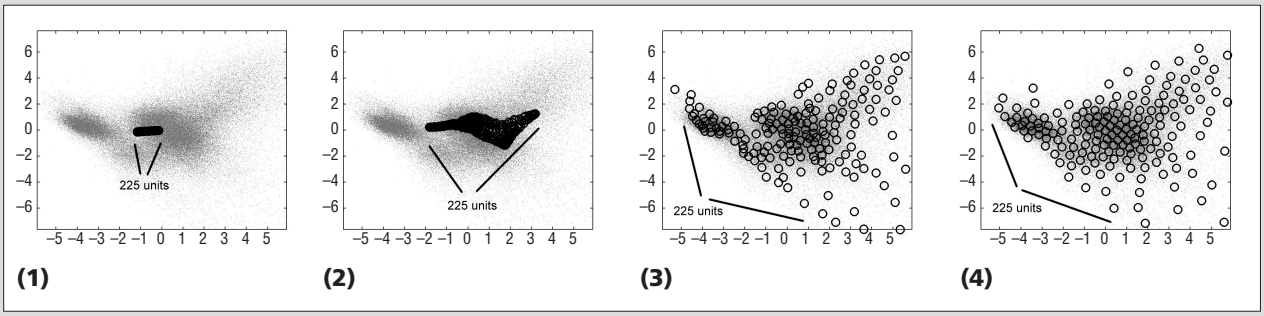

Figure A. The static neural gas model (NG) after (1) 0, (2) 20,000, (3) 200,000, and (4) 400,000 iterations. NG starts from 15×15 units with a small random weight and spreads the gas during its convergence.

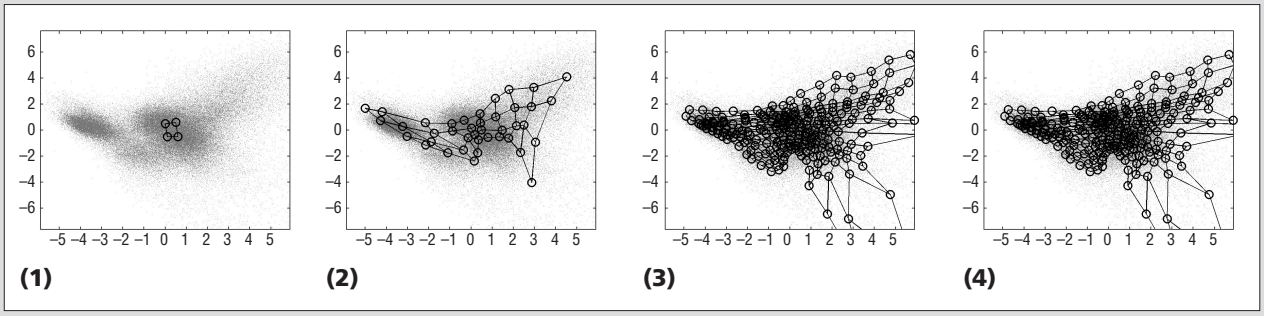

Figure B. The dynamic neural model (growing grid) after (1) 0, (2) 20,000, (3) 200,000, and (4) 400,000 iterations. GG starts from a grid of 2×2 units and grows a row or a column of units.

ranges from 75.25 and 81.72 percent for 10,000 full-text documents and AQE ranges from 2.538 to 3.511. Table 3 shows our results.

Integrating top-down knowledge from WordNet in all six algorithms achieves much better performance for two evaluation criteria. CA increased from 15.86 to 19.59 percent, with accuracy increasing from 94.84 to 97.58 percent. AQE drops from 5.68 to 10.44 percent and has a lower value—between 2.273 and 3.186. We also notice that a higher CA is associated with a lower

- The GCS dynamic neural model keeps its units with a triangle connectivity, as Figure C shows. It starts from three units; splitting the farthest unit from the unit with the biggest error inserts a new unit. GCS removes units with very low probability density (meaning few input vectors are mapped to it) with their direct neighbors in the corresponding triangle.
- The GNG neural model applies the GCS growth mechanism for the competitive Hebbian learning topology.[9] GNG starts from two units and connects an input sample's BMU to the second match unit as direct neighbors, as Figure D shows. Splitting the unit with the highest error in the direct neighborhood from the unit with the highest error in the entire structure inserts a new unit. Units are pruned if their connections are not strong enough. Both GCS and GNG have two learning rates: one for BMU and the other for BMU's direct neighbors.

References

1. S. Grossberg, "Competitive Learning: From Interactive Activation to Adaptive Resonance," *Cognitive Science*, vol. 11, 1987, pp. 23–63.
2. T. Kohonen, "Self-Organized Formation of Topologically Correct Feature Maps," *Biological Cybernetics*, vol. 43, 1982, pp. 59–69.
3. T. Martinetz and K. Schulten, "A 'Neural-Gas' Network Learns Topologies," *Artificial Neural Network,* vol. I, 1991, pp. 397–402.
4. J. MacQueen, "Some Method for Classification and Analysis of Multivariate Observations," *Proc. 5th Berkeley Symp. Math. Statistics and Probability,* Univ. of California Press, 1967, pp. 281–297.
5. B. Fritzke, "Kohonen Feature Maps and Growing Cell Structures: A Performance Comparison," *Neural Information Processing Systems 5*, C.L. Giles, S.J Hanson, and J.D. Cowan, eds., Morgan Kaufmann, 1993.
6. B. Fritzke, "Growing Grid: A Self-Organizing Network with Constant Neighborhood Range and Adaptation Strength," *Neural Processing Letters*, vol. 2, no. 5, 1995, pp. 9–13.
7. B. Fritzke, "Growing Cell Structures: A Self-Organizing Network for Unsupervised and Supervised Learning," *Neural Networks*, vol. 7, no. 9, 1994, pp. 1441–1460.
8. B. Fritzke, "A Growing Neural Gas Network Learns Topologies," *Advances in Neural Information Processing Systems 7*, G. Tesauro, D.S. Touretzky, and T.K. Leen, eds., MIT Press, pp. 625–632.
9. T.M. Martinetz, "Competitive Hebbian Learning Rule Forms Perfectly Topology Preserving Maps," *Proc. Int'l Conf. Artificial Neural Networks* (ICANN 93), Springer-Verlag, 1993, pp. 427–434.

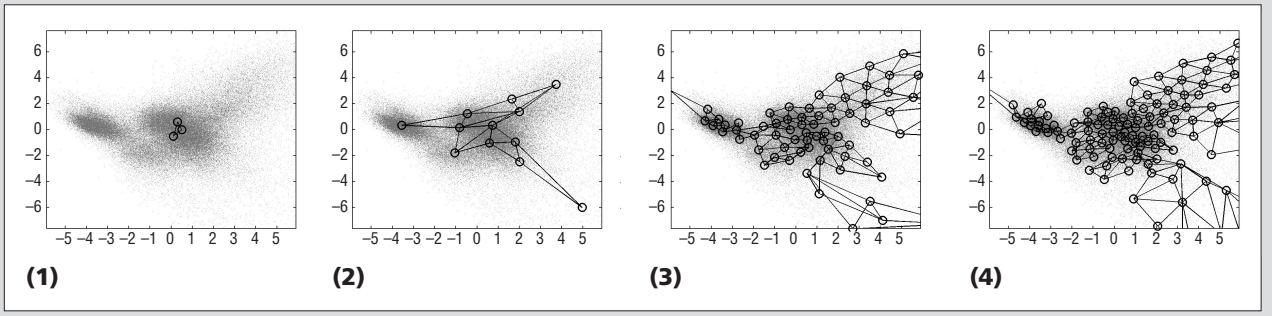

Figure C. The dynamic neural model (growing cell structure) after (1) 0, (2) 20,000, (3) 200,000, and (4) 400,000 iterations. GCS starts from a triangular structure of three units and grows by a single unit at a time, always keeping its triangular structure.

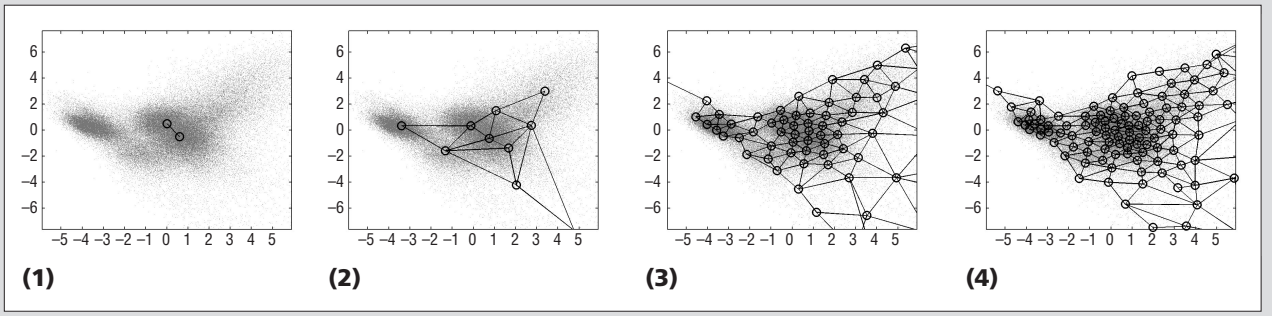

Figure D. The dynamic neural model (growing neural gas) after (1) 0, (2) 20,000, (3) 200,000, and (4) 400,000 iterations. GNG starts from a two-unit structure and grows a single unit at a time. GNG maintains its structure by connecting the best matching unit and the second BMU of an input vector.

AQE. That is, VQ performance is a significant feature in clustering. The smaller the AQE, the more cohesive the cluster; the more cohesive the cluster, the more accurate the model.

From a network architecture viewpoint, we can redivide the six competitive models into three groups:

- SOM and GG have a constraint on their positional units, which always form a grid architecture. However, the true relationship between data vectors differs from the

Table 3. Classification accuracy (CA) and average quantization error (AQE) of six competitive methods for 10,000 full-text news articles. We use one-level hypernym for systems integrated with WordNet.

	CA (percent)			AQE		
Model	With WordNet	Without WordNet	Improvement	With WordNet	Without WordNet	Improvement (percent)
CL	81.72	97.58	15.86	2.538	2.273	10.44
SOM	78.05	95.80	17.75	3.511	3.183	9.34
NG	80.68	96.84	16.16	2.722	2.485	8.71
GG	75.25	94.84	19.59	3.378	3.186	5.68
GCS	79.12	95.82	16.70	2.824	2.585	8.46
GNG	78.80	95.87	17.07	2.799	2.550	8.75

Table 4. Classification accuracy (in percent) of our hybrid model using different WordNet hypernym levels for 10,000 full-text news articles.

Model	One-level hypernym	Improvement	Two-level hypernym	Improvement	Three-level hypernym	Improvement
CL	97.58	15.86	94.61	12.89	90.04	8.32
SOM	95.80	17.75	91.87	13.82	81.55	3.50
NG	96.84	16.16	93.90	13.22	89.62	8.94
GG	94.84	19.59	89.19	13.95	84.39	9.14
GCS	95.82	16.70	92.58	13.46	86.94	7.82
GNG	95.87	17.07	92.50	13.70	87.50	8.70

Table 5. AQE of our hybrid model using different WordNet hypernym levels for 10,000 full-text news articles.

Model	One-level hypernym	Improvement (percent)	Two-level hypernym	Improvement (percent)	Three-level hypernym	Improvement (percent)
CL	2.273	10.44	2.666	–5.04	2.768	–9.06
SOM	3.183	9.34	3.598	–2.48	3.752	–6.86
NG	2.485	8.71	2.848	–4.63	2.993	–9.96
GG	3.186	5.68	3.557	–2.27	3.617	–4.00
GCS	2.585	8.46	2.965	–4.99	3.057	–8.25
GNG	2.550	8.75	2.940	–5.04	3.024	–8.04

artificial grid-based relationship. This might be one reason why the AQEs are higher than those in other models.

- GNG and GCS also make some assumptions about the relationship between units. However, the unit growing and pruning feature allows for adjusting the model while learning. Thus, these models' AQEs are smaller than those in the SOM and GG group.
- NG and CL make no presumption about the relationship between units. This group has the lowest AQEs.

Varying hypernym levels. The higher the hypernym level, the more general the concept meaning. If the level is too high, different senses of a word will be treated as the same word, which isn't what we want. We find that the one-level hypernym suits our model, as Tables 4 and 5 show.

Scaling up the experiment. We've scaled up our experiment to use a 100,000 full-text training set. The results are similar to those achieved using a 10,000 full-text data set.

Integrating top-down knowledge from WordNet in all six algorithms based on two evaluation criteria resulted in much better performance. Using these models alone gives us a CA between 71.56 and 73.06 percent and an AQE between 3.580 and 4.470, as Table 6 shows. Our hybrid approach achieves an improvement in CA between 21.36 and 31.46 percent and between 93.02 and 95.33 percent accuracy. AQE decreases between 13.45 and 15.82 percent and has a smaller value between 3.070 and 3.782. Thus, our model can potentially handle real-world tasks.

Our current hybrid model does not intend to identify each word's meaning, but rather considers each word as a symbol. The model therefore uses traditional VSM representation techniques, such as TFxIDF, which supposes that all words are mutually independent and the sequences of words in sentences are ignored. Two documents with similar words are represented by similar weight vectors and thus are located in a neighborhood in the multidimensional space. However, identifying the true meaning of a word or document can reduce the redundancy of similar words. It is possible to

Table 6. Classification accuracy (CA) and average quantization error (AQE) of six competitive methods for 100,000 full-text news articles.

	CA (percent)			AQE		
Model	Without WordNet	With WordNet	Improvement	Without WordNet	With WordNet	Improvement (percent)
CL	72.19	94.90	31.46	3.580	3.070	14.25
SOM	72.08	93.25	29.37	4.470	3.782	15.39
NG	73.06	95.33	22.27	3.657	3.165	13.45
GG	71.56	93.02	21.46	4.179	3.518	15.82
GCS	72.91	94.27	21.36	3.696	3.187	13.77
GNG	72.80	94.47	21.67	3.704	3.197	13.69

propose a new vector representation technique using the word concept instead of the statistical word knowledge. Therefore, integrating some natural language processing techniques, such as tagging, parsing, and word sense disambiguation with our hybrid model can further reduce the gap between human classification and data-driven neural clustering. ■

Acknowledgments

We thank Michael Oakes for comments on an earlier draft of this article.

References

1. T. Kohonen, "Self-Organized Formation of Topologically Correct Feature Maps," *Biological Cybernetics*, vol. 43, 1982, pp. 59–69.
2. T. Kohonen et al., "Self Organization of a Massive Document Collection," *IEEE Trans. Neural Networks*, vol. 11, no. 3, May 2000, pp. 574–585.
3. T. Kohonen et al., "Very Large Two-Level SOM for the Browsing of Newsgroups," *Proc. Int'l Conf. Artificial Neural Networks* (ICANN 96), LNCS 1112, Springer-Verlag, 1996, pp. 269–274.
4. C. Hung and S. Wermter, "A Dynamic Adaptive Self-Organizing Hybrid Model for Text Clustering," *Proc. 3rd IEEE Int'l Conf. Data Mining* (ICDM 03), IEEE Press, 2003, pp. 75–82.
5. T. Kohonen, *Self-Organizing Maps*, Springer-Verlag, 2001.
6. G. Salton, *Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer*, Addison-Wesley, 1989.
7. H. Chen, C. Schuffels, and R. Orwig, "Internet Categorization and Search: A Self-Organizing Approach," *J. Visual Comm. and Image Representation*, vol. 7, no. 1, Mar. 1996, pp. 88–102.
8. S. Wermter, *Hybrid Connectionist Natural Language Processing*, Neural Computing Series, Chapman & Hall, 1995.
9. G.A. Miller, "WordNet: A Dictionary Browser," *Proc. 1st Int'l Conf. Information in Data*, 1985, pp. 25–28.
10. D. Brezeale, *The Organization of Internet Web Pages Using WordNet and Self-Organizing Maps,* master's thesis, Univ. of Texas at Arlington, 1999.
11. S. Scott and S. Matwin, "Text Classification Using WordNet Hypernyms," *Proc. COLING/ACL Workshop on Usage of WordNet in Natural Language Processing Systems*, Morgan Kaufmann, 1998, pp. 38–44.
12. M. de Buenaga, J.M. Gómez-Hidalgo, and B. Díaz-Agudo, "Using WordNet to Complement Training Information in Text Categorization," *Recent Advances in Natural Language Processing*, 1997, pp. 150–157.
13. M. Riedmiller and H. Braun, "A Direct Adaptive Method for Faster Backpropagation Learning: The RPROP Algorithm," *Proc. IEEE Intl. Conf. Neural Networks* (ICNN 93), IEEE Press, 1993, pp. 586–591.

The Authors

Chihli Hung is a PhD researcher in the Center for Hybrid Intelligent Systems at the University of Sunderland School of Computing and Technology and a lecturer at the De Lin Institute of Technology, Taiwan. His research interests include intelligence systems, artificial neural networks, cognitive neuroscience, natural language processing, and electronic commerce. He has an MSc (Distinction) in computer science from the University of Sunderland. Contact him at the Centre for Hybrid Intelligent Systems, School of Computing and Technology, Univ. of Sunderland, St. Peters Way, Sunderland SR6 0DD, UK; chihli.hung@sunderland.ac.uk.

Stefan Wermter is a professor in Intelligent Systems and leads the Centre for Hybrid Intelligent Systems at the University of Sunderland. His research interests include artificial intelligence, neural networks, cognitive neuroscience, hybrid systems, language processing, and learning robots. He has an MSc from the University of Massachusetts and a PhD from the University of Hamburg, both in computer science. Contact him at the Centre for Hybrid Intelligent Systems, School of Computing and Technology, Univ. of Sunderland, St. Peters Way, Sunderland SR6 0DD, UK; stefan.wermter@sunderland.ac.uk; www.his.sunderland.ac.uk.

Peter Smith is dean of the School of Computing and Technology at the University of Sunderland. His research interests include expert systems, knowledge engineering, and computers in manufacturing. He is particularly interested in developing novel techniques to solve real business and industrial problems. He has a BSc (Hons) in mathematics and computing and a PhD in computer simulation from the University of Sunderland. He is a Fellow of the British Computer Society, a Chartered Engineer, a Chartered Mathematician, and a Fellow of the Institute of Mathematics and its Applications. Contact him at the School of Computing and Technology, Univ. of Sunderland, St. Peters Way, Sunderland SR6 0DD, UK; peter.smith@sunderland.ac.uk.