

White Paper:

Implementing a Categorisation Engine (SOM)

Version 1.0 (30th July 2003)

Graham Whitehouse

In2itive Business Group

An FSDC plc Group Company

Registered Office: 15 The Metro Centre, Welbeck Way, Peterborough PE2 7UH
Registered in England No: 4132210

Implementing Categorisation Engines for Knowledge Management

Background

This is a technical white paper that looks at the reasons for using categorisation, and covers, in some detail, the methods and techniques used in the In2itive Categorisation engine.

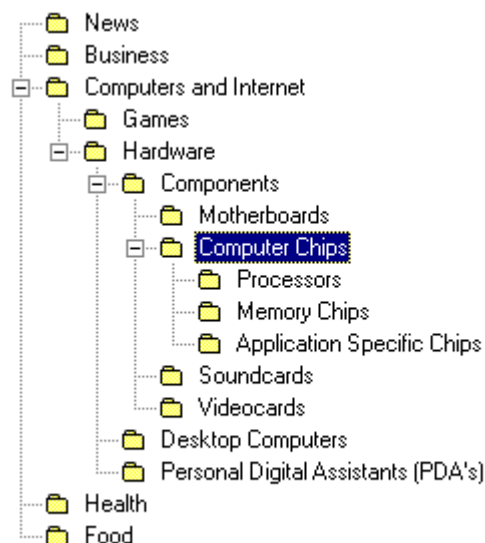
Taxonomy and Content Classification

It has been recognised that one limiting factor of information retrieval technologies is the high recall and low precision of results returned. In other words, it is not uncommon to get thousands or millions of results returned from a query. This situation is exacerbated by the fact that users typically only use one or two words to define their information need and rarely use advanced query techniques to improve the meaning of the query.

For example, a search for 'chips' using the Google search engine returns some 5,500,000 hits, including some about potato chips, casino chips and many about computer chips. There are many different types of computer chips, for example processor chips, memory chips and application specific chips but all are returned in the same list, so finding documents relevant to a search is made difficult. Query based retrieval is useful when the user is familiar with the subject, and the vocabulary of a domain is known. In many domains however, where the language is constantly evolving as well as the links between subject areas, it can result in wasted time performing a multitude of searches.

Browsing through a classification structure can be used as an alternative to searching for information. It is much easier to discover information about a particular subject if you see it in the context of related information.

Given the 'chips' example, consider the hierarchical taxonomy shown in below:



In taxonomy such as this, information is grouped into topics, with related content being placed in the same topic. The taxonomy is represented using 'isa' and 'superset' relations (and their inverses 'member' and 'subset'). 'isa' links an object to a set of which it is a member, and 'superset' links a set to a superset of it (thus 'member' links a set to one of its members, and 'subset' links a set to one of its subsets). As an example, a 'dogs' category would be a subset of an 'animals' category and the 'animals' category may be the superset of a 'cats' category.

A label that describes the main themes in each topic is used to represent the topic, for example 'Computers and Internet', 'Hardware', 'Components' and 'Microprocessors'. By making a selection at each level of the hierarchy the degree of ambiguity is further reduced until the most relevant topic (in this case 'Computer Chips') is found.

Taxonomy software correlates and groups unstructured information from a myriad of sources. Taxonomy can be considered as computer generated card catalogues that allow us to locate, retrieve and cross-reference information.

A problem with the taxonomy approach is that as it grows, or when people first start to use it, it can be difficult to know where a particular sub-category may be. For example, is music a sub-category of arts or entertainment? Over time, as people become familiar with the classification scheme this becomes less of a problem.

Document Classification

A great deal of explicit organisational knowledge is contained within documents. Document classification techniques are used to generate and/or populate taxonomies of documents, emails and any other text. Manual methods involve human effort in creating the structure and classifying documents by hand, as would a traditional librarian. Although manual classification can be logical and accurate, it has the drawback of being very labour-intensive and costly. Automatic methods are defined as either supervised or unsupervised.

Supervised Document Classification

A system that performs supervised document classification aims to label natural language texts with thematic categories from a predefined set. The categories are just symbolic labels and no additional knowledge of their meaning is available.

The fact that the categories are manually defined in supervised document classification is both an advantage and a disadvantage. For some industries a well-defined taxonomy already exists.

This is true in certain organisations, where employees are used to classifying information by organisational function or department. One disadvantage is that it can take a great deal of time and effort constructing the taxonomy and training the classifier – in some domains there could be many thousands of categories and sub-categories. Another disadvantage is that individuals tend to have different views of the information in an organisation, and how different pieces of information are related so individual bias can be involved in deciding on the categories.

Unsupervised Document Classification

Cluster analysis is a statistical technique that seeks to identify groups, or clusters, of similar objects in a multi-dimensional space. Cluster analysis of documents (or document clustering), aims to automatically group documents, or sections of documents, into clusters that have similar contents. In contrast to supervised methods, no predefined categories exist and the clustering algorithm creates categories based on the clusters generated.

Document clustering has mainly been based on the Cluster Hypothesis: 'Closely associated documents tend to be relevant to the same requests'. The idea is that the relevant documents are more similar to each other than to non-relevant documents. If this hypothesis holds on a particular collection then retrieval should be improved because the class or category, once found will contain only the relevant documents.

Another way of using document clustering is to give the user the ability to browse through the classification structure, exploring different areas in the collection. This is especially useful when the user has difficulty expressing their information need. The user may not be looking for anything specific but may just wish to explore the general database contents with the vague aim of finding something interesting.

Document clustering has also been used in the organising of results returned by a search engine in response to a users query (post-retrieval document clustering). The results can be grouped into categories to help the user in finding relevant documents.

Clustering Techniques

There are a number of analysis methods have been used for document clustering, two are detailed below:

Partitional techniques create a one-level partitioning of the data points, however the weaknesses of this technique are well known. First of all, the number of clusters has to be pre-specified, and as the number of clusters grows, for example to thousands of clusters, it becomes untenable.

Hierarchical techniques produce a treelike construction where clusters of closely related documents are nested within bigger clusters containing a set of documents that are less similar. There are two basic approaches to generating a hierarchical clustering agglomerative and divisive.

- 1) **Agglomerative** algorithms find clusters by initially assigning each object to its own cluster and then repeatedly merging pairs of clusters until a certain stopping criteria is met.
- 2) **Divisive** algorithms start with one, all-inclusive cluster and, at each step, split the cluster until only singleton clusters of individual points remain. In this case, we need to decide, at each step, with cluster to split and how to perform the split.

Self Organising Maps

Introduction to Artificial Neural Networks

An artificial neural network (ANN) is an information-processing system that has certain performance characteristics in common with biological neural networks. Many tasks that seem simple for us, such as reading a handwritten note or recognizing a face, are difficult for even the most advanced computer. In an effort to increase the computer's ability to perform such tasks, ANNs have been used to allow computers to act more like the human brain, with its neurons and synaptic connections.

A neural network consists of a large number of connected processing elements known as neurons. Each neuron is linked to certain of its neighbours with varying coefficients of connectivity, known as weights that represent the strengths of these connections.

Learning is accomplished by adjusting weights to cause the overall network to output appropriate results; this process is known as **training**.

Supervised Training versus Unsupervised Training

The training method is an important distinguishing characteristic of a neural network. Supervised training involves presenting a sequence of training vectors, each with an associated target output vector. The weights are then adjusted according to the learning algorithm.

Unsupervised training involves neural networks grouping similar input vectors together without the use of training data to specify what a typical member of each group looks like. Input vectors are presented to the network but no target output vectors are specified. The network will modify the weights so that the most similar input vectors are assigned to the same output neuron (or cluster). Neural networks that use unsupervised training are also known as 'self-organising'.

The Self Organising Map

The Self-Organising Map (SOM) is a popular unsupervised neural network that was first introduced by *Kohonen*. With a SOM, data from a high-dimensional input space is mapped onto a low (mainly two) dimensional output space. The SOM algorithm has been particularly successful because of its topological preservation i.e. the structure of the input data is preserved as faithfully as possible in the output.

Pre-Processing

Electronic documents cannot be applied to a computer algorithm in their natural state; some form of processing is required to put them into a structure that can be used by the algorithm. The most common statistical method of document

representation is the Vector Space Model (VSM). The basic VSM involves storing documents as vectors in which each element corresponds to the frequency of a term in the document (usually normalised to a value between 0 and 1).

Essentially this model provides a 'bag of words' in that the positioning of each term is ignored. Usually the words are weighted according to their power of discrimination between topics, i.e. words that have limited discrimination power need to be de-emphasised and vice-versa.

A common approach to weighting is term frequency (TF) multiplied by inverse document frequency (IDF). TF is the number of times a term appears in a document and IDF is the inverse of the frequency of the term in the document collection. This gives a higher weighting to words in documents that appear less frequently in the document collection. Those terms that appear frequently over the whole document collection are poor discriminators and so are given lower weightings using this formula.

When using individual words as features, the semantics or meanings of the words are lost as it is the surrounding words that provide context. To overcome this problem, a succession of words in a short context is used. This method creates a larger set of features from which to select the most relevant. It is argued that a single unit of meaning is represented by phrases such as 'foreign exchange market', whereas by using each word alone ambiguity is introduced. The downside of this method is that it increases the number of dimensions of the document vectors. With the previous example, 'foreign', 'exchange', 'market', 'foreign exchange', 'exchange market' and 'foreign exchange market' provides six dimensions instead of three.

Feature Set Reduction Techniques

'The Curse of Dimensionality', coined by *Bellman*, relates to a number of problems arising from increased data dimensions: essentially the amount of data to sustain a given spatial density increases exponentially with the dimensionality of the input space. Samples quickly become lost in the wealth of space when dimensionality becomes too large, with points tending to become equidistant from one another. All problems become tougher as the dimensionality increases. Nowhere is this more prevalent than in problems related to high dimensional information spaces, such as a large document collection. As a result of this, attempts are made to reduce dimensions where possible.

Feature Selection

The first step in reducing the dimensionality of the entire document collection is by feature selection. Words that are believed to contain little or no meaning can be removed from the list of candidates. These terms make what is commonly referred to as a stop list and typically contain prepositions, pronouns, articles and other non-descriptive words, for example, 'the', 'at' or 'into'.

Word Stemming

In natural language processing, conflation is the process of merging or grouping together non-identical words that refer to the same principal concept. A word-stemming algorithm can be used to conflate terms, for example, by removing any attached suffixes (for example '-ly', '-ness' or '-ment') and in some cases prefixes (such as 'anti-', 'bi-', or 'semi-') from terms. This can be a useful tool in dimensionality reduction since the stem of a term represents a broader concept than the original term.

For example 'employing', 'employs' and 'employed' have the same stem 'employ'. By applying a stemming algorithm three dimensions are reduced to one. In its simplest form such a transformation is the conversion of a plural to singular form.

Term Frequency Thresholds

Luhn first suggested that pre-set thresholds could be used to remove terms appearing in more than an upper bound and less and a lower bound number of documents. Luhn stated that the ability of words to discriminate content reached a peak at a rank order position half way between the two cut offs positions and fell off to near zero, from the peak in both directions, when nearing the cut off points.

Rare words as well as very common words are assumed to contain very little information. Hence, it is safe to remove them and thus significantly reduce the dimensionality of the input space without compromising the text classification performance of the system.

Training

A SOM implementation has an output layer of interconnected neurons that are fully connected to the input layer. This connection has a weight as described earlier. Each neuron has an associated weight vector with an equal number of weights as the input vector, with each representing a particular feature in the domain being modelled.

In the core SOM algorithm, there are two main steps. These are calculating the winning neuron, and updating the weights on the winning and neighbouring neurons.

The neighbourhood function is a time decreasing function, which determines to what extent the neighbours of the winning unit are updated.

Labelling

In order to interpret the trained SOM, the clusters need to be labelled with a meaningful piece of text. Labelling methods aim to find the features that best represent the data associated with a particular cluster.

For example if all of the documents associated with a node concerned business, then 'business' would be a suitable label for the cluster. To avoid having to perform this operation manually, automatic methods are used.

Hierarchical Feature Maps

The idea of hierarchical feature maps is to apply a hierarchical arrangement of several layers containing two-dimensional SOM. For each output unit in one layer in the hierarchy, a two-dimensional SOM is added to the next layer. Only the documents assigned to the unit's node are used for training the corresponding child map. The training of each single SOM follows the basic SOM algorithm.

Summary

When large amounts of information is being used, applying computational power to the problems can, if properly considered, provide real business advantages.

In this case, extensive research into existing and new techniques has brought about a categorisation engine (or SOM) that can be used to create a taxonomy from a wide range of documents.

Inclusion of advanced linguistic functions that were developed at the same time offer a unique combination to the market.

Where do In2itive fit in?

In2itive started a project just over two years ago to build a knowledge management solution that could both be used to extend their portal offerings as well as sold as a stand-alone solution.

After some extensive research, they decided to work in close co-operation with the University of Manchester Institute of Science and Technology (UMIST), as they believed that great value could be derived from use of both advanced linguistics and advanced categorisation techniques in a Knowledge Management engine.

Many of the items described in this paper were actually developed during this period, and indeed, patents have now been granted to In2itive on the way that the categorisation is used in this environment.

Utilising theorists and experts as well as quality programmers meant that foundations of the linguistic algorithms are of excellent breeding, and they are within the control of the group for building future upgrades and developments.