

A Neural Network based Software Retrieval System with Fuzzy-Related Thesaurus

Huilin Ye

School of Electrical Engineering and Computer Science
The University of Newcastle, Callaghan, NSW 2308, Australia
Huilin.Ye@newcastle.edu.au

Abstract: The qualities of both the classification and retrieval queries have significant impacts on the retrieval performance of a software retrieval system. A classification scheme based on a Nested Self-Organising Map (NSOM) and a query refinement mechanism based on a fuzzy-related thesaurus were proposed to promote the qualities. An NSOM consists of a top map and a set of nested maps. The retrieval on the top map maintains high recall while the retrieval on the nested maps enhances precision. A fuzzy-related thesaurus can be generated from an NSOM. The user can reformulate an improved query by adding terms or replacing an original query term with an appropriate term stored in the thesaurus. The experimental results reveal that both the NSOM and query refinement significantly improved the retrieval performance.

Keywords: Self-organising map, Software retrieval, Query refinement, Thesaurus.

I. Introduction

The performance of a retrieval system is usually measured by *recall* and *precision*. Recall is the proportion of relevant material retrieved, measuring how well a system retrieves all the relevant material. Precision is the proportion of retrieved material that is relevant, measuring how well the system retrieves *only* the relevant material. Recall and precision tend to be related inversely. When a search is broadened to achieve better recall, precision tends to go down and vice versa. In the context of software retrieval, for a given query we want to find the most relevant software component for reusing it with minimum effort to adapt it to a new application. In this case, precision is more important than recall. However, if the recall is not maintained in a reasonable level some most relevant software components may be missed. Therefore, how to improve precision without excessive compromise of recall is crucial for software retrieval systems. To achieve such a retrieval performance an appropriate classification scheme must be developed.

Another factor that will significantly influence the retrieval performance is the quality of retrieval queries. Formulating precise and effective queries in information

retrieval systems has always been a difficult task, even for experienced users [1]. When searching for information to solve a problem, people often do not have a clear idea of what information is needed. Searching for information may be regarded as a situation of irresolution or an anomalous state of knowledge, in which users believe that the knowledge that can help solve their problem exists, but they are unable to characterise the problem or articulate their information needs adequately. Therefore, ill-defined queries are very common in retrieval systems and a query refinement mechanism is necessary to help promote retrieval performance.

There has been a large amount of effort devoted to finding suitable approaches to building software retrieval systems [2-4]. However, it was concluded by Mili et al. [5] this issue has not been satisfactorily solved. This paper will present a neural network based software retrieval system with a fuzzy-related thesaurus to enhance retrieval precision without excessively compromising recall. Both the classification scheme and the fuzzy-related thesaurus based query refinement are based on an unsupervised neural network, the Self-Organising Map (SOM) [6].

SOM has been extensively used in document classification [7-9]. Such classifications are usually coarse-grained and cannot accommodate high precision in information retrieval [8]. We developed a sophisticated neural network architecture, called Nested Self-Organising Map (NSOM), to achieve an optimal balance between recall and precision. The NSOM based classification will be done in two levels and the accuracy of the classification will be enhanced from the first coarse-grained level to the second fine-grained level. The coarse-grained classification at the first level is used to maintain a high level of recall and the precision will be improved by the fine-grained classifications at the second level.

Query refinement is an essential information retrieval tool that interactively recommends new terms related to a particular query for the user to improve the quality of the query. When the user interacts with a retrieval system, the system provides term excerpts that are considered relevant to

a particular user query. The user can then reformulate the query by adding terms from the excerpts or replacing an original query term with a related term stored in the excerpts. This is called *thesaurus-based query refinement*. A fuzzy-related thesaurus is used in this retrieval system.

Several retrieval experiments have been done and very promising results have been observed. The NSOM based retrieval improved precision in comparison with other retrieval systems. The query refinement worked well for the ill-defined queries and a significant improvement on both recall and precision was obtained.

The remainder of the paper is organised as follows. Section II presents an overview of the retrieval system. Section III describes how to represent a natural language query for the retrieval. Section IV presents a sample NSOM based retrieval session to show how the retrieval system works. Section V discusses the query refinement mechanism. The experimental results are presented in Section VI. Section VII concludes the paper.

II. Overview of the Software Retrieval Systems

This software retrieval system is based on software textual documentation associated with software components rather than on the components themselves. The most significant characteristic of software components is their functionality for the purpose of reuse. Most software components contain textual descriptions of their functionality in the form of system descriptions, operation manuals or user documents etc. Because the major interest in software retrieval is the functionality of the required components, these natural language documents can be used as the surrogates of the software components in the process of software classification and retrieval.

The system consists of a number of modules, the representation scheme using automatic indexing approach to transform the software documents into feature vectors, the NSOM based classification scheme classifying the feature vectors, and the retrieval mechanism with a fuzzy-related thesaurus.

The representation scheme uses an automatic free-text indexing method to identify the features associated with each component. An automatic indexing method is used to identify the indices associated with a software document collection, called a corpus. Each index can be considered as a feature belonging to the document from which it is identified. The total number of the indices obtained from a corpus is the dimension of the feature vector. The feature space containing a number of feature vectors will be presented to the SOM as its input data.

A SOM can learn from its input data. Each input stimulus elicits a localised response. This corresponds to a non-linear projection of the input data onto the network that makes the most important semantic relationships among the input data items geometrically explicit [6]. It is this property of SOM that makes it useful for classification. An NSOM consists of

a number of SOMs and they are organised in two levels, a single map at the top level and a number of nested maps at the second level. Each nested map contains a subset of the original document collection. A coarse-grained classification on the top map will support high recall and the fine-grained classifications on the nested maps will accommodate high precision.

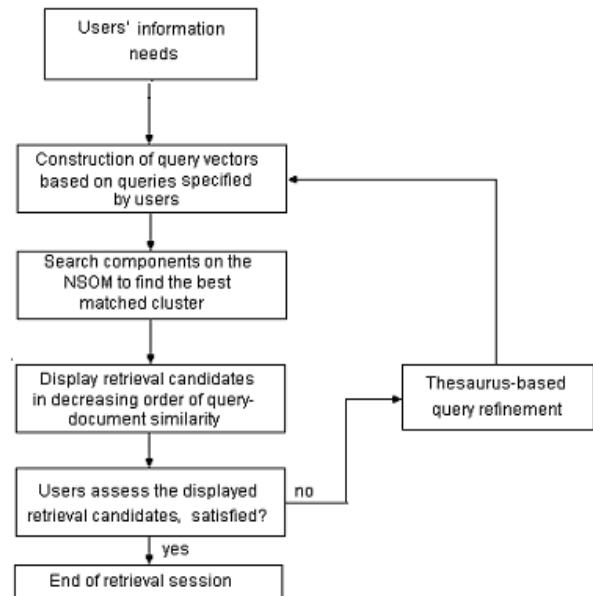


Figure 1. Retrieval process

The retrieval mechanism enables the user to express queries in natural language without the need of understanding the inner working of the retrieval mechanism. A domain dependent fuzzy-related thesaurus is developed for query refinement to help improve the retrieval performance for ill-defined queries. The details of the representation and classification schemes have been reported in [10-11]. This paper will concentrate on the retrieval mechanism. A typical retrieval process is presented in Figure 1.

III. Query Representation

The first step in retrieval is the formulation of queries. Traditionally, queries are specified by the user according to an authorised formalism. It is usually required that the user should have good knowledge about the formalism and the inner working of the software library, which costs much user effort spent in formulating queries. This system accepts natural language queries to minimise the user effort in query formulation. The queries will undergo an indexing process first, and then will be represented as query vectors based on the indexing results.

The query indexing includes deleting stop words, stemming, replacing single terms with concept classes, and phrase formation. The specific methods used in the indexing process are the same as used in software document indexing

and have been described in detail in [11]. As a result of indexing, queries are represented by a set of features. Assume that there is an n -dimensional document feature space to which a query will be mapped. The query will also be represented by an n -dimensional vector $q_i = [\sigma_{i1}, \sigma_{i2}, \dots, \sigma_{ij}, \dots, \sigma_{in}]^T$. Each element of the query vector corresponds to the presence or absence of a certain feature in the document feature space. If all the features assigned to a query are included in the document feature space, then the query vector will be formed easily by simply assigning the weights of these features to the corresponding elements and assigning zero to all the other elements in the query vector. Unfortunately, this is not always the case. The following steps may be used to deal with query features that are not included in the document feature space:

Step 1: A domain-dependent dictionary is employed to find a synonymous document feature for a given query feature.

Step 2: Query features that fail to find synonymous document features in Step 1 will be discarded because these features are considered irrelevant to the components stored in the system.

The formed query vector can then be submitted to the retrieval system and the system will search the desired component(s) by projecting the query vector onto a pre-established NSOM to find the potential retrieval candidates.

IV. NSOM-based Retrieval

The architecture of the NSOM is shown in Figure 2. It consists of a top map (TM) and a number of nested maps (NM) (only one example is shown). The TM contains a whole software component collection and the NMs are software component maps of the sub-collections of the whole set of components. Assuming there is a certain sub-area of the TM whose centroid is node c (the hexagon enclosed by dotted lines in Figure 2), there will be a number of documents mapped within this sub-area. The number of features associated with this sub document collection is much smaller than the full collection because of the small size of the sub-collection. Therefore, the sub-collection can be represented by a set of feature vectors with a much lower dimension. These feature vectors will be used to train a NM. On completion of the training, a more elaborate map of the sub-collection will be formed on the NM. The detailed algorithm of how to divide the top map into a number of NMs has been presented in [10].

This architecture enables a two-step retrieval process. The first step of the retrieval process conducts a coarse-grained retrieval on the TM while the second step performs a fine-grained retrieval on the NMs. First, the query vector will be mapped onto the TM. The location of the query vector on the TM will determine the corresponding NM. The query vector will then be mapped onto the NM and the matched cluster will be found on the NM. The components located in the matched cluster will be ranked according to their

similarity to the query. The ranked component list will be returned to the user as the retrieval candidates.

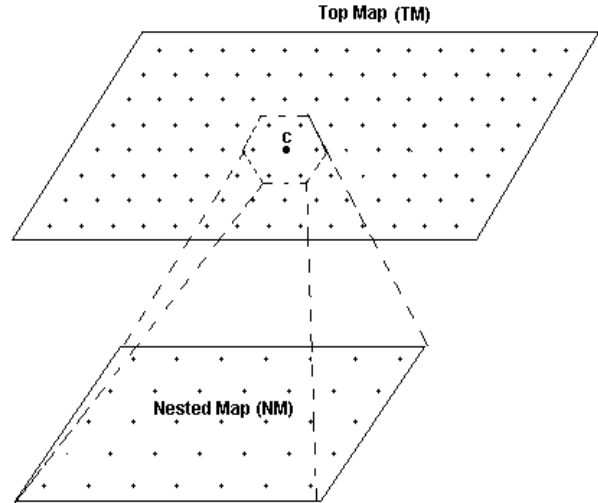


Figure 2. Architecture of the NSOM

A sample retrieval session presented here is based on an NSOM containing all the manual pages in the first section of the Unix User Manual. The TM is shown in Figure 3 where the number associated with each neuron indicates how many components reside at the neuron. Assume that a user query “How to transfer files over Internet?” is issued and the corresponding query vector is projected onto a node, called winning node (marked as “c” in Figure 3), on the TM. The matched cluster is the small hexagon plotted in solid lines in Figure 3 where 14 components reside. The retrieval targets for the query are `ftp` and `tftp` which are included in the cluster. If the retrieval procedure stops here, a high recall (1.00) and a poor precision (0.14) will be obtained.

A corresponding NM (shown in Figure 4) is constructed based on a sub-collection containing 53 components located in the area enclosed by the dotted lines in Figure 3. The original query will then be mapped onto the NM and a matched cluster enclosed by the dotted lines in Figure 4 is found. Components located in the cluster are `tftp`, `telnet` and `ftp`. The retrieval recall is 1 and precision is 0.666. Comparing this result with the result achieved at the TM, a significant improvement of precision is observed.

The enhanced retrieval performance achieved by the NSOM is under an assumption that the user queries are defined adequately. However, as discussed earlier, formulating precise and effective queries has always been a difficult task and ill-defined queries will have negative impact on the retrieval performance. A query refinement mechanism is provided to help the user overcome this difficulty.

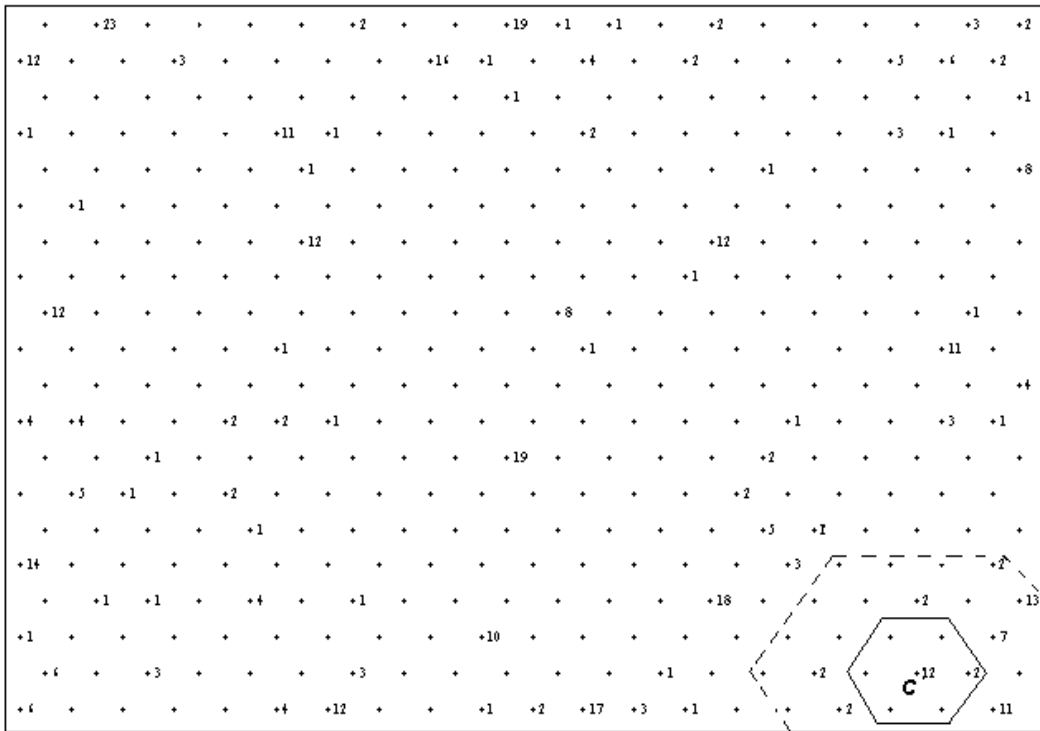


Figure 3. A TM containing a Unix component collection

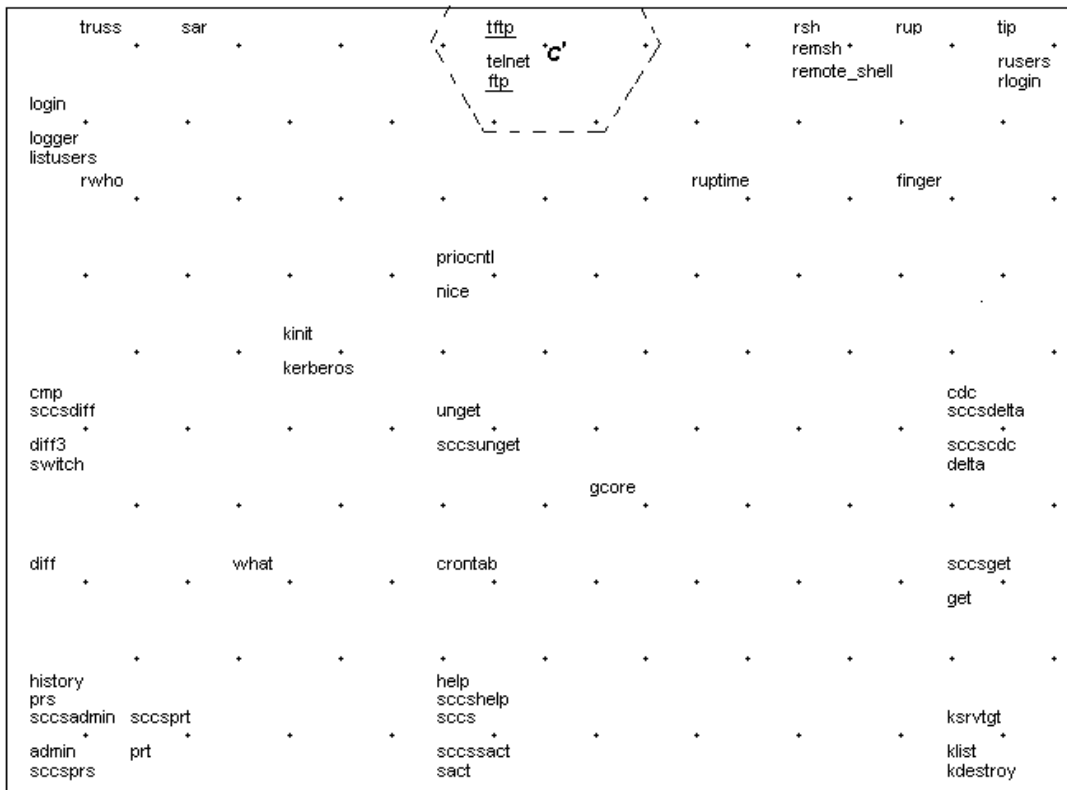


Figure 4. An NM derived from the TM shown in Figure 3.

V. Query Refinement with a Fuzzy-Related Thesaurus

For any given query feature thesaurus-based refinement will provide temporary feature excerpts in which the stored words or phrases have various kinds of relationships with the query feature. A relational thesaurus is usually used for this purpose [12]. However, a relational thesaurus contains limited feature relationships, such as *synonyms*, *generic-specific*, and *whole-part*, among the terms. In reality the relationships among the features are more complex than those are identified in the thesaurus. Explicitly identifying these complex feature relationships is a difficult job. But for the purpose of the query refinement for retrieval if different groups of features can be found and each group of features contribute to characterise a certain cluster these features can be considered close related. Assume that the user issues an ill-defined query containing only a few features that are relevant to the desired cluster. If the user can expand the query with the features in the feature group that characterise the desired cluster it is very likely the desired cluster will be targeted by the refined query. Fortunately, we can identify these feature groups using the trained SOMs. Software components located in a certain cluster on a map have similar functionality. This functionality are characterised by the most highly weighted features associated with the cluster. In other words, highly weighted features existing in a cluster describe similar concepts. Thus, these features can be considered as fuzzy-related.

For a given query feature, its fuzzy-related features contained in an NSOM can be dynamically obtained. A given query feature can be mapped onto the TM and a winning node will be identified. The top 20 highly weighted features associated with the cluster are eligible for collecting into a fuzzy-related thesaurus. These highly weighted features will be provided to the user for expanding the query or replacing an original feature with a fuzzy-related new feature to improve the quality of the query.

VI. Retrieval Experiments

Several retrieval experiments have been done to assess the performance of the retrieval system. The measures used in the experiments are recall and precision. The hypotheses to be tested in the experiments are:

1. The NSOM based classification can enhance retrieval performance in comparison with other retrieval systems.
2. The thesaurus based query refinement can improve the retrieval results for ill-defined queries.
3. Incorporating the query refinement into an NSOM based retrieval system will further enhance the retrieval performance for ill-defined queries.

The first experiment for testing hypothesis 1 was intended to assess the effectiveness of the NSOM based

classification only without involving the query refinement mechanism. The NSOM used in the experiment is the one presented in Section IV that contains all the manual pages in the first section of the Unix User Manual. First, the NSOM was compared with Guru, which is a software retrieval system considered capable of achieving a better-than-average retrieval performance [13]. Then, NSOM was compared with a publicly available full-text retrieval system—Personal Librarian (PL). It was observed that this system achieved an improvement of 4.59% on recall and 16.60% on precision in comparison with Guru, and an improvement of 35.87% on recall and 52.24% on precision in comparison with PL. The details of the experiment have been reported in [10].

A second experiment for testing hypothesis 2 was intended to test the fuzzy-related thesaurus based query refinement only without involving NSOM based classification. The thesaurus has been applied to a small collection containing only 97 Unix manual pages classified on a single SOM and promising results have been reported in [14].

This paper will present the third experiment that focuses on the hypothesis 3. The retrievals will be done in two different procedures. One procedure will use thesaurus based query refinement but the other will not use the query refinement. The retrieval results obtained from the two procedures will be compared. Salton [15] stated three requirements that any representative test procedure must satisfy:

1. The queries, used for test purposes, must be user search requests actually submitted to and processed by the system.
2. The test collection must consist of documents originally included in the library, chosen in such a way that any advance knowledge concerning the retrievability of any given component by either system is effectively ignored.
3. The number of components of retrieval candidates selected by the two systems must correspond to the same cut-off.

The first requirement was satisfied because the query set used in the experiment was collected from the users. The author conducted a survey from several Unix users at Southern Cross University, Australia, and collected a number of queries for retrieving Unix manual pages. Among the collected queries, 14 poorly defined queries were selected for the experiment.

For the second requirement, we used the test collection corresponding exactly to Section 1 of the Unix User Manual, i.e. the NSOM presented in Section IV. No advance knowledge has been used when choosing the documents but simply used the whole collection of the Section 1 of the Unix manuals.

As far as the third requirement is concerned, a retrieval cut-off in NSOM-based retrieval is determined by a cluster distance, *dc*. When a query is projected onto a map, the winning node and the cluster distance will specify a certain

area on the map. Components located in such an area will be selected as retrieval candidates. For example, the area enclosed by the dotted lines in Figure 4 is the cluster found for the query “How to transfer files over Internet?”. The winning node for the query is marked as c and the cluster distance is 1. As retrieval precision is much more crucial than retrieval recall in software retrievals, a small cluster will achieve higher precision than a big cluster. The same cluster distance $dc = 1$ is chosen for both retrieval procedures. The retrieval results achieved by the two procedures can then be compared.

The experiment consists of the following steps:

- The queries were submitted to the system without any query refinement. The average recall and precision of the retrieval results were obtained.
- The queries were given to 3 subjects randomly selected from 6 volunteers and they were asked to use the thesaurus to refine the queries and submit them to the system to get the retrieval results.
- For each query, average recall and average precision based on the three subjects’ retrieval results were calculated.
- An overall average recall and precision for all the queries were calculated. The retrieval performance is compared in Table 1.

Retrieval	Average recall	Average precision
normal queries	0.866	0.743
ill-defined queries without query refinement	0.573	0.362
ill-defined queries with the thesaurus based refinement	0.756	0.632

Table 1. Comparison of retrieval results

The retrieval result for normal queries shown in Table 1 was obtained in the first experiment. Comparing this result with the retrieval result of the ill-defined queries (without query refinement) a large decline of both recall and precision in the ill-defined queries was observed. This means the quality of the queries has significant impact on the retrieval performance. For the same ill-defined query set the retrieval results after the fuzzy-related thesaurus based query refinement have been improved substantially. The recall and precision are improved by 32% and 75% respectively.

The three subjects endorsed that the thesaurus based query refinement is capable of enlightening them to choose appropriate terms for the refinement. The thesaurus provides intuitive information about the relevant terms for a given query to help the user find out the desired terms if the user suffers from the experience knows as “I cannot explain what I want, but I’ll recognize it if I see it”.

VII. Conclusions

In this paper, the NSOM based software retrieval system with a fuzzy-related thesaurus based query refinement is discussed. The problem of coarse-grained classification occurring in the previous SOM-based applications is isolated on the TM and these clusters can be fine-grained on the NMs. As a result, the NSOM based retrieval can achieve better retrieval performance. The retrieval on the TM maintains a high level of recall, and the retrieval on the NMs enhances precision. A trained NSOM classifies not only the software components but also the features associated with the components. Based on the classified features a fuzzy-related thesaurus can be generated to accommodate query refinements. The user can reformulate a query by adding terms to or replacing a query term from the original query with an appropriate term stored in the thesaurus.

Experimental results were compared and discussed. The results reveal that the NSOM based retrieval enhanced retrieval performance in comparison with Guru. Guru’s retrieval performance was believed to be more than satisfactory and better than the average information retrieval systems [13]. To test the effectiveness of the fuzzy-related thesaurus based query refinement, retrieval results with query refinements and without query refinement for the same set of ill-defined queries are collected. It was observed that the retrieval system was capable of achieving a much more effective retrieval performance using the fuzzy-related thesaurus based query refinement. The improvements of the recall and precision achieved by the query refinement are 32% and 75% respectively.

References

- [1] B. Velez, R. Weiss, M. Sheldon and D. Gifford. “Fast and Effective Query Refinement”. In *Proceedings of the 20th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 6-15, 1997.
- [2] R. Prieto-Diaz. “Implementing Faceted Classification for Software Reuse”, *Communications of the ACM*, 34(5), pp. 88-97, 1991.
- [3] Y. Maarek, D. Berry and G. Kaiser. “An Information Retrieval Approach for Automatically Construction of Software Libraries”, *IEEE Transactions on Software Engineering*, 17 (8), pp. 800-813.
- [4] T. Isakowitz and R. Kauffman. “Supporting Search for Reusable Software Object”, *IEEE Transactions on Software Engineering*, 22 (6), pp. 407-423, 1992.
- [5] A. Mili, R. Mili and R. Mittermeir. “A Survey of Software Reuse Libraries”, *Annals of Software Engineering*, 5, pp. 349-414, 1998.
- [6] T. Kohonen. *Self-Organising Maps*, Springer-Verlag, Berlin, 1997.
- [7] T. Kohonen. “Self-Organisation of Very Large Document Collections: State of the Art”. In *Proceedings of the 8th International Conference on*

- Artificial Neural Networks*, Springer, Skovde, Sweden pp. 55-74, 1998.
- [8] R. Orwig, H. Chen and J. Nunamaker. "A Graphical, Self-Organising Approach to Classifying Electronic Meeting Output", *Journal of the American Society for Information Science*, 48 (2), pp. 157-170, 1997.
- [9] X. Lin, D. Soergel and G. Marchionini. "A Self-Organising Semantic Map for Information Retrieval". In *Proceedings of the 14th Annual International ACM/SIGIR Conference on Research and Development in Information Retrieval*, Chicago, pp. 262-269, 1991.
- [10] H. Ye. "A Self-Organized Software Library". In *Neural Networks Applications in Information Technology and Web Engineering*, Borneo Publishing, 2005.
- [11] H. Ye, and B. W. N. Lo. "Toward a Self-Structuring Software Library", *IEE Proceedings – Software*, 148 (2) pp. 45-55, 2001.
- [12] G. Salton. *Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer*, Addison-Wesley, 1989.
- [13] Y. Maarek. "Software Library Construction from an IR Perspective", *SIGIR Forum*, 25 (2), pp. 8-18, 1991.
- [14] H. Ye and H. Liu. "A Fuzzy-Related Thesaurus for Query Refinement", *Neural Processing Letters*, 19 (2), pp. 97-107, 2004.
- [15] G. Salton. "Recent Studies in Automatic Text Analysis and Document Retrieval.", *JACM*, 20 (2), pp. 258-278, 1973.

Author Biography

Dr Huilin Ye is a Senior Lecturer at the School of Electrical Engineering and Computer Science, the University of Newcastle, Australia. She received a BEng at Harbin Engineering University, China in 1982 and a PhD in Software Engineering at Southern Cross University, Australia in 2001. Her research interests include software classification and retrieval, neural network applications, feature modelling in software product lines, and data mining.