Augmenting Information Retrieval by Knowledge Infusion

Jeff Riley
Hewlett-Packard Australia
31-41 Joseph St.
Blackburn 3130
Australia
jeff@aus.hp.com

Kemal A. Delic
Hewlett-Packard France
5, Avenue Raymond Chanas - Eybens
38053 Grenoble
France
kemal_delic@hp.com

Keywords: Help Desk, Information Retrieval, Knowledge Infusion, Genetic Algorithm, Neuro-Fuzzy

Abstract

WiseWare is an operational knowledge-based system providing support for help desk analysts, users in Information Technology departments and channel partners [Delic and Lahaix, 1998]. A WiseWare document is an entity containing human readable knowledge expressed in free form English text. Documents are evaluated and annotated for complexity, quality, usefulness and age, etc. The retrieval of documents relevant to some query can be augmented by the infusion of some domain knowledge. This knowledge infusion can take place at one or both of two distinct places in the retrieval process:

- before the search, knowledge of a user's habits, preferences and query history, etc. can be used to modify a query

- after the search, knowledge of a user's preferences, skill level and viewing history, etc. can be combined with knowledge of the retrieved documents' complexity, quality, age and source, etc. to rank and sort the results in a more relevant manner.

This paper investigates the benefit of knowledge infusion after the search process.

1 Introduction

The vast proliferation of desktop computers in industry and business is generating an ever increasing need for effective and efficient internal, enterprise help desks. In the past, and to some extent currently, help desks have been staffed by highly skilled and knowledgeable technicians, or experts. Calls to the help desks usually describe some defect, fault, or deviation in expected behaviour; or more generally describe some problem encountered by the end user. Industry and business are deploying larger, more complex and heterogenous systems, with end users increasingly more remote from the central systems. This trend, along with the explosive proliferation of both small and large systems, is contributing to a sharp increase in the volume and complexity of calls to help desks and software support centres. As the volume and complexity of calls to these centres increases, the cost of employing more expert technicians becomes prohibitively expensive. The trend is for somewhat less knowledgeable and experienced technicians to handle the more common, less complex problems; allowing the expert, and expensive, technicians to deal with the less common, more complex problems. Either way, current software support strategies are very technician intensive, so methods of improving productivity and reducing costs by automating areas of the help desk environment are being sought.

Eighty-five per cent of problems dealt with by help desks are useability issues; ten per cent are administrative issues; and just five per cent are software defects or undocumented features [Rose, 1998]. Of the eighty-five per cent of problems which are useability issues, sixty per cent could have been resolved by the user reading the documentation [Rose, 1998]. It is accepted that in the help desk environment problem recognition and assessment account for eighty percent of the time involved, while problem solving or routing accounts for only twenty percent. Reducing the amount of time spent on the recognition and assessment of problems, both those which are highly repetitive and have simple, well-known solutions and those less frequently seen and which require more complex problem solving techniques, will have a significant impact on problem resolution time, resolution rate, cost of resolution and ultimately customer satisfaction.

Help desk systems are usually composed of two main components: the help desk front-end and the help desk back-end. The front-end typically manages the help desk technician's interaction with the end user or customer. For example, the customer's name and contact details will be recorded and accessible via the help desk front-end, as will other details such as the problem description and on-going problem resolution documentation. The help desk back-end on the other hand deals with the technician's knowledge access (Figure 1).

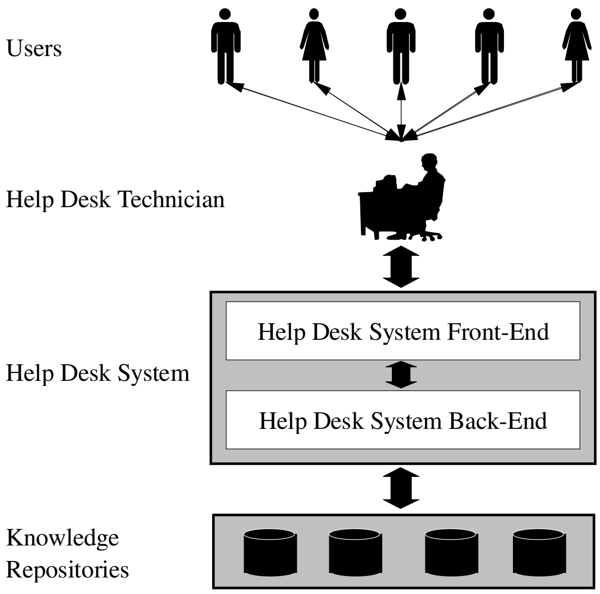

Figure 1 Conceptual Help Desk System.

Support technicians solve problems sometimes by the application of some innate knowledge, but more typically by matching a problem with similar problems either in their own previous experience, or some (usually electronic) knowledge repository. Support knowledge contained within the knowledge repository is usually some formatted or stylised text document(s) describing the problem and the solution offered or conclusion reached. Automatically matching a problem with similar problems recorded in such a knowledge repository and providing the support technician with trusted solutions is a desirable method of reducing the amount of time, and therefore cost, spent on problem recognition and assessment. This can be stated more succinctly as the task of finding documents within a changing electronic corpus which satisfy some information need, or query. Current efforts attempt to retrieve knowledge contained in documents by the use of index terms, key words and key phrases. This is an inherently imprecise method yielding less than satisfactory results. The aim of this work is to develop a (hybrid) method using knowledge infusion which yields better results.

2 Method Description

2.1 Overview

This work investigates the effectiveness of a hybrid Information Retrieval method involving the use of a genetic algorithm [Holland, 1975], a neuro-fuzzy classifier, and the infusion of some domain knowledge. In this hybrid method, the genetic algorithm is used to determine, by simulated evolution, the set of key phrases which best describes each of the different categories of documents. In this implementation, key phrases are an extension of key words: a key phrase consists of one or more (possibly separated) words. Once the key phrases which are considered to best describe each category are determined, a neuro-fuzzy classifier is used to assess the contents of and classify documents according to the appearance of the key phrases. More precisely, vectors of key phrases which describe (possibly multiple) documents are assessed and classified. A neuro-fuzzy classifier is proposed so that fuzzy rules may more easily be extracted from the resultant system. Once extracted, the fuzzy rules can be applied to the classification task more efficiently than a neural network, and extra rules describing the domain knowledge may be added, thus augmenting the classification and retrieval process.

The document assessment and classification system is comprised of a parser, a key phrase extraction unit, a genetic algorithm, and a neuro-fuzzy classifier (Figure 2).

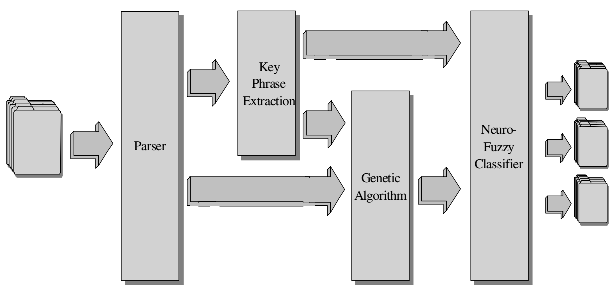

Figure 2 Assessment and classification system overview.

2.2 Parser and Key Phrase Extraction

Since not all words or phrases contained within the text of a document are considered useful for content assessment and recognition, only a subset of the phrases from any document are passed through to the genetic algorithm and classifier for analysis. The parser first strips all stopwords from the document. Stopwords are those words which are considered not to be semantically significant and generally occur with high frequency (ie. *a, as, is, it, the*, etc.).

The format of some documents, or sections of some documents (for example, header information), may allow the parser to pass to the genetic algorithm and classifier the words found in the document or section of the document in their entirety. Otherwise all phrases considered key phrases are extracted from the document and only those key phrases are passed to the genetic algorithm and classifier.

The initial implementation of the system uses a master list of key phrases. For efficiency reasons the master list is limited in the initial implementation to an arbitrary maximum number of phrases. The master list is generated by a group of experts who choose the phrases considered most useful as key phrases based upon technical meaning, frequency of occurrence in the corpus etc.

2.3 Genetic Algorithm

In this genetic model the genes are key phrases, and chromosomes are documents. The documents are reduced to vectors of key phrases by the parser and represented by bit-strings. In the initial implementation, each bit in the bit-string represents a key phrase, with the value of the bit ("*1*" or "*0*") indicating the presence or absence of the key phrase in the document. The population of competing representations is modified over time by the genetic operators, with the goal being to evolve the set of key phrases which best describes the collection of documents making up a particular category. This is similar to [Gordon, 1988]. The genetic operators implemented for this model are crossover (initially single point) and mutation (single bit).

The absence of one key phrase may be just as important as the presence of another in determining the classification of a document. For this reason the vector of key phrases representing a document presented to the genetic algorithm must contain an indication of the presence or absence of every key phrase in the master list of keywords.

The initial implementation of the system encodes only absence or presence information onto the chromosome presented to the genetic algorithm. The chromosome presented to the genetic algorithm therefore consists of a number of bits equal to the maximum number of key phrases, with each bit representing the presence or absence of a key phrase.

In later implementations, a chromosome based on the in-document frequency of occurrence of key phrases will be evaluated. This would increase the size of the chromosome by some factor depending upon the number of bits used to represent the frequency of occurrence, hence increasing the complexity and execution speed. Since the determination of the optimum set of key phrases for classification is an offline or batch task, speed of execution is not considered critical.

2.4 Neuro-Fuzzy Classifier

The neuro-fuzzy classifier implemented in this work is based on the neuro-fuzzy classifier proposed by Jang in [Jang, 1992]. The general architecture for the neuro-fuzzy classifier is shown in Figure 3.

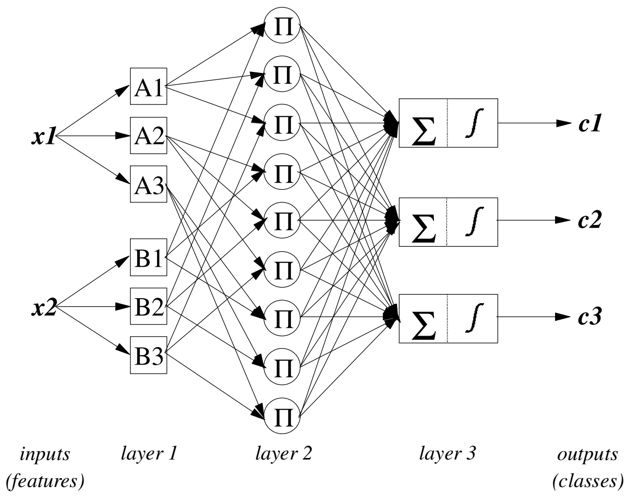

Figure 3 Neuro-Fuzzy Classifier Architecture.

The presence of a key phrase in the text of a document is considered to be, at least in part, a measure of the strength with which the document is related to the topic indicated by the key phrase. The text of any document may (probably will) include key phrases which indicate different topics. For each key phrase in the master list, input to the neuro-fuzzy classifier is a function of the *document length* and the key phrase's *collection frequency* and *term frequency*. This is similar to the *relevance weight* of [Robertson *et al.*, 1995] and [Robertson and Sparck Jones, 1997], and is considered to be a measure of the strength of the relationship between the key phrase and the document content. This input is fuzzified to represent the degree of membership of three fuzzy subsets representing the strength of the relationship: *low*, *medium* and *high*.

The output of each output node of the neuro-fuzzy classifier as shown in Figure 3 is the degree to which the presented pattern belongs to the category corresponding to the output node. This output is post-processed to produce either a normalised fuzzy output or a crisp output (Figure 4).

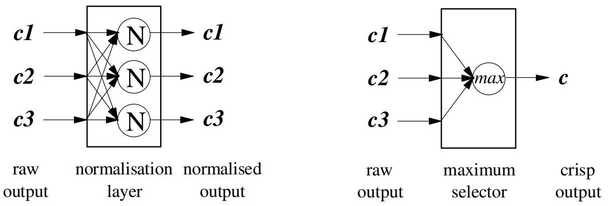

Figure 4 Fuzzy normalisation layer and crisp output selector.

For the classification task the classified documents are tagged with the normalised fuzzy output for each category reflecting the reality that documents could be considered to contain multiple concepts, or be related to several categories. The fuzzy output indicates the degree to which the document is related to each category. If it is required that each document be classified to a single category, the crisp post-processing layer could be used. The training task for this classifier benefits from the fact that prior knowledge about the training data set can be directly onto the classifier's parameters [Jang, 1992]. This allows the training process to begin from a good initial point, so training speed is improved. The parameters learned from the training process are then transformed into fuzzy **if-then** rules for the classification process, and domain knowledge can be infused at this point by adding fuzzy rules developed from the domain knowledge.

2.5 Knowledge Infusion

The general knowledge infusion process proposed is shown in Figure 5. This paper deals with the infusion of knowledge after the search process only. Future work will include query modification by the use of domain knowledge before the search. Knowledge of the document collection and the user's past interactions and preferences, in the form of fuzzy **if-then** rules, is injected after the extraction of the rules from the neuro-fuzzy classifier. While the rules generated by the neuro-fuzzy classifier use information contained within the documents being classified, the addition of rules containing domain knowledge augments the classification and retrieval process by utilising knowledge which is not otherwise apparent. The additional knowledge used to augment the classification and retrieval process is knowledge of the document source, quality, age, and complexity etc.; as well as knowledge of the user's skill level and viewing history etc.

Since the goal of this information retrieval process is to return documents to the user which contain trusted solution(s) relevant to a problem expressed as the query, considering knowledge of the quality of the document is an obvious refinement of the search process. The use of other domain knowledge may be somewhat less obvious. Knowing for example, from skill level and viewing history, that a user discards documents of a certain perceived complexity, we may choose to rank those documents lower in the results list. Care must be taken with this approach, as documents containing trusted, relevant solutions must still be presented to the user. Considering this additional knowledge while ranking and classifying the retrieved documents aids in presenting documents to the user which are most relevant to that user's needs.

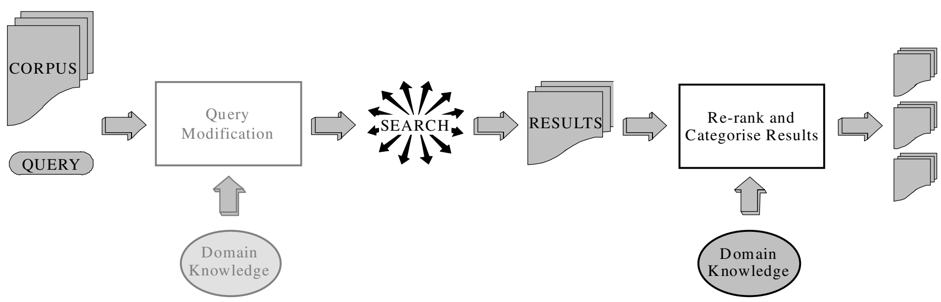

Figure 5 Knowledge infusion process.

The hypothesis that the application of domain knowledge will improve the retrieval process will be tested by the measurement of the retrieval process both with and without the infusion of domain knowledge. A sample of 1000 documents will be extracted from the WiseWare knowledge repository with each document being manually tagged by expert users for complexity, quality, etc. Benchmark tests will be conducted with no domain knowledge infusion, then tests conducted with domain knowledge infusion after the search.

3 Conclusions

We believe that Information Retrieval by the use of keywords or phrases has reached certain technological limitations and that, at least in a help desk environment, the use of certain domain knowledge can improve the retrieval process. Our hypothesis, in the framework of the genetic algorithm and neuro-fuzzy classifier described, will be tested on a sample collection from a live, operational help desk system and the results presented.

References

[Delic and Lahaix, 1998] Delic, K., and Lahaix, D. Knowledge Harvesting, Articulation, and Delivery. *HP Journal, May 1998:74-81.*

[Gordon, 1988] Gordon, M. Probabilistic and Genetic Algorithms for Document Retrieval. In *Communications of the ACM, Vol. 31 No. 10.*

[Holland, 1975] Holland, J. Adaptation in Natural and Artificial Systems. *Ann Arbor: The University of Michigan Press.*

[Jang, 1992] Jang, J-S. Neuro-Fuzzy Modeling: Architectures, Analyses and Applications. *PhD thesis, Department of Electrical Engineering and Computer Science, University of California, Berkeley, California, USA.*

[Robertson *et al.*, 1995] Robertson, S., Walker, S., and Hancock-Beaulieu, M. Large Text Collection Experiments on an Operational, Interactive System: Okapi at TREC. In *Information Processing and Management, Vol. 31 No. 3, 1995.*

[Robertson and Sparck Jones, 1997] Robertson, S., and Sparck Jones, K. Simple, Proven Approaches to Text Retrieval. *Technical Report 356, Computer Laboratory, University of Cambridge.*

[Rose, 1998] Rose, B. The Future of Software Support. In *Transcript from STARS Software Support Conference, 1998.*