3-2004

Intelligent Query Answering Through Rule Learning and Generalization

James M. Carsten

**INTELLIGENT QUERY ANSWERING
THROUGH RULE LEARNING AND GENERALIZATION**

THESIS

James M. Carsten, Captain, USAF

AFIT/GCS/ENG/04-05

**DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY**

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT/GCS/ENG/04-05

INTELLIGENT QUERY ANSWERING THROUGH RULE LEARNING AND GENERALIZATION

THESIS

Presented to the Faculty

Department of Electrical and Computer Engineering

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

In Partial Fulfillment of the Requirements for the

Degree of Master of Science in Computer Systems

James M. Carsten, BS

Captain, USAF

March 2004

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

Acknowledgments

I would like to express my sincere appreciation to my thesis advisor, Dr. Bert Peterson, for his guidance, support and most of all patience throughout the course of this thesis effort. While I was very much unprepared for the AI sequence, he showed great restraint and diligence while he dragged me through the entire process.

I also thank the most interesting professor I've ever met, Dr. Potoczny, who convinced me that brute force is a viable approach to a master's degree. His humor kept us interested, and his tests kept us busy (for days.) Dr. Potoczny is on my thesis committee, as is Dr. Baldwin (thanks for the help with my statistical analysis, Maj. B.)

My hat is also tipped to the A-Team, for keeping my box full of useless, but often funny, e-mail. Special thanks to my study partner, Flt Lt Wendell Fox of the Australian Air Force. Other people that made a difference are Ken Laviers, my AI lab partner and Java support desk. I can't forget the Halo Death Squad: Squatch, Rage, Cable Guy, Serial, Snap, Flynt, Switch and anyone else who butt-stroked[1] me from behind. The hours we spent decompressing helped me stay sane. Most of these guys also got me through a class or two.

To the men and women of Skydive Green County in Xenia: You guys were my saving grace on the weekends. Without the freedom of the blue skies and your companionship at 150mph+, this place would have been total misery. I will miss you all and will come back to visit. Blue Skies!

And finally, I dedicate my effort to my parents, who gave me the love and encouragement to get through AFIT and on to Tinker (where I'll be the closest to home since 1983.)

Jamie Carsten

[1]In the game of Halo, a butt stroke is hitting someone with the butt of your current weapon.

Table of Contents

List of Figures

List of Tables

AFIT/GCS/ENG/04-05

Abstract

The Department of Defense (DoD) relies heavily on information systems to complete a myriad of tasks, from day-to-day personnel actions to mission critical imagery retrieval, intelligence analysis, and mission planning. The astronomical growth in size and performance of data storage systems leads to problems in processing the amount of data returned on any given query. Typical relational database systems return a set of unordered records. This approach is acceptable in small information systems, but in large systems, such as military image retrieval systems with more than 1 million records, it requires considerable time (often hours to days) to sort through thousands of records and select the relevant for analysis.

This research introduces Intelligent Query Answering (IQA) as a novel approach to information retrieval. IQA implements the FOIL algorithm to learn rules based upon user feedback [QUI90]. The Winnow algorithm adjusts rule weights based on user classification, for improved document orderings [BLU97]. A semantic tree specific to the domain allows rule generalization across the domain.

Testing shows a document sort accuracy rate of 63-93% against a controlled test dataset and 78-89% accuracy rate on a subset of declassified National Air Intelligence Center imagery metadata. These results demonstrate that this research provides groundwork for future efforts in rule learning and rule generalization in the information retrieval field.

INTELLIGENT QUERY ANSWERING
THROUGH RULE LEARNING AND GENERALIZATION

1. Introduction

1.1 Purpose

In today's world most organizations rely heavily on information and information technology to conduct day-to-day activities. Recent events in the war against terrorism illustrate the critical need for real-time, accurate intelligence information. The ability of the Department of Defense and the Air Force to accomplish their mission relies heavily on the ability to process a tremendous amount of data, both text and imagery for intelligence analysis.

Over the years, millions of records have been collected, cataloged, digitized, and stored in large databases. Data storage systems are continually expanding to meet the ever-increasing demand for more capacity. It is common to find a personal computer with 40-120 gigabytes of hard disk storage. Large computer systems measure storage in terms of terabytes (1 terabyte = 1024 gigabytes), and now systems are even entering the 2-petabyte range of capacity (1 petabyte = 1,024 terabytes) [XIN03].

As storage capacity increases, the computational cost of manipulating that information also increases. The available information is overwhelming to even the most accomplished information processing organizations. This problem becomes more pronounced in systems with heterogeneous data collections. Returning a set of hundreds or thousands of unordered records dramatically increases the time spent sorting through the data to find the desired information. To be an effective tool for users, computer systems must have more sophisticated ways of returning relevant information to user queries.

This research introduces Intelligent Query Answering (IQA) as a novel approach to information retrieval. IQA uses a modified version of Quinlan's FOIL algorithm to learn rules based upon user search terms and classification of returned documents [QUI90]. The Winnow algorithm adjusts the rule weights based on previous user classifications, improves the order of the sorted documents returned, and the process repeats [BLU97]. A semantic tree specific to the domain allows rule generalization. This provides users with documents sorted with the assistance of generalized rules where none previously existed, and also generalizes similar sets of specialized rules

1.2 Background

Several research efforts at the Air Force Institute of Technology have focused on improving user access to relevant information with the National Air Intelligence Center's (NAIC) the Imagery Exploitation Capability (IEC) System. The IEC System is an operational system in need of improvement. Some of these efforts have included improving the methods of returning relevant information by using multi-modal feedback [WIL03] and by improving the graphical user interface [BAC03]. These efforts have made significant strides in ordering records returned by relevance in user queries, but require an extensive number of queries on the IEC System to build effective structures that improve search results and overall query performance. The need to develop better methods for providing relevant information faster is clear.

1.2.1 NAIC IEC System Background

The NAIC uses the IEC System to store and retrieve images used for intelligence analysis and planning. This system has been in use four years and consists of more than 1.3 million images with associated metadata [BAC03]. It consists of a database and an image library. The database stores the metadata for each image and has a hyperlink. The metadata in each record describes the image while the hyperlink points at the respective image in the image library. The goal of the system is the retrieval of military images in support of timely intelligence analysis. Although a relatively new software product, IEC has an extraordinarily slow response time (minutes) and returns unordered sets of records, 5 records at a time. Most of a researcher's time is spent waiting for IEC responses.

1.2.2 IEC Operations

The NAIC employs more than 700 personnel who use the IEC system. A person assumes one of four specific roles using this system: photographer, commenter, researcher, or analyst [BAK03, DIA03]. Photographers are responsible for acquiring imagery. Commenters digitize the imagery and store them in the IEC System. They also add comments (metadata) to the system that describe an image. The images are stored in the imagery library and the metadata is stored in the relational database. Researchers receive requests for specific image content and search the system for images that assist the requesting analyst. One or more query terms are used to search for relevant images, much like one would use an Internet search engine. A researcher makes note of any relevant images and passes that information to an analyst. Analysts review these images and provide analysis for the intelligence community.

1.2.3 IEC Issues

The IEC system is an enormous relational database. Each record contains a link to the respective image it represents in the image library. It responds to a researcher's query by providing a complete, unordered list of records containing only documents that include all query search terms in the metadata. Furthermore, these queries cannot be Boolean.

Boolean searches use the logical operators *and*, *not* and *or*. The Boolean *and* means that all the terms specified must appear in the document(s). The Boolean *or* means that at least one of the terms specified must appear in the document(s). The Boolean *not* means that at least one of the terms you specify must not appear in the document(s). Combinations of these terms can provide an effective return of documents albeit without regard to relevance.

Since the IEC does not have Boolean search capability, a user may not search by *and*-ing, *or*-ing, or *not*-ing terms together to increase the effective return of records. All query terms must have a matching term in the each record's metadata (effectively all terms *and*-ed together) for the IEC to return the record. Additionally, term order has no relevance in the IEC.

IEC returns five records at a time and the time delay for the appearance of the first set of records is usually greater than 30 seconds and often as long as eight minutes

[DIA03]. Within each record returned is also a hyperlink for the image associated with the metadata. In order for the researcher to view an image, they must click on this hyperlink and retrieve the image. The time delay between the researcher selecting the image and the image appearing can be as long as two minutes [DIA03]. The time to change from one set of five records to the following five takes from two to five minutes [DIA03]. This delay occurs each time the researcher requests a new set of five records. The IEC with its 1.3 million images frequently returns hundreds of unordered records. Occasionally a query results in more than a thousand records returned. This makes the task of finding relevant images tedious and time consuming, with individual searches taking hours or days to complete. Given the number of records routinely returned, there is a substantial possibility that the researcher will never see records deep in the returned list.

Other approaches using modern information retrieval methods to improve the IEC system capabilities have been studied. These approaches have been somewhat successful, but the basis for this research is the exploration of an alternative method of returning relevant records using machine learning techniques. The IEC provides a useful source of data for study. Section 2.2 presents an overview of some information retrieval methods to provide a contrast for the basis of this research.

1.3 Research Focus

The primary focus of this research is the introduction and exploration of a new method of information retrieval that blends rule learning through user search and document classification with rule generalization. Learning rules through user classification provides the basis for returning records sorted by relevance. Generalizing those learned rules across a predefined semantic tree provides a “best guess” return of relevant documents based upon existing rules. The goal is a system that rapidly learns how a user queries a database, and then uses those rules to return the most relevant documents.

1.3.1 Objectives

This research has two objectives. The first objective is the identification and implementation of an effective rule learning system, including user feedback and relevance assignment. Rule learning and rule weight adjusting add relevance to each document. This provides a method of returning documents in order of relevance. The second objective is defining a data structure to represent the semantic relationship of a dataset. This structure would support term generalization and and allow for interrogation of that data structure. WordNet [MIL90] provides some ideas for generalizing terms. Rule generalization adds additional relevance to documents and improves relevance order. It also reduces computation time by combining two or more specialized rules in to a more general one.

1.3.2 Approach

This research approach begins with a review of published literature on information retrieval, rule learning and lexicographical dictionaries. It continues with the selection and implementation of a suitable rule-learning method. An electronic lexicographical dictionary guides the building of a generalization framework. Nouns and adjectives from the IEC System’s metadata form the generalization hierarchy. This hierarchy provides the foundation for rule generalization. Experiments generate and generalize rules through user queries. An analysis of rules learned and document return order determines the effectiveness of the combined methodologies.

1.4 Summary

The primary focus of this research is the introduction and exploration of a new method for information retrieval. This research presents and implements a methodology for blending rule learning with rule generalization for improved query results. Test and result analyses validate the approach and provide a way of quantify its successes. This research uses test data and the de-classified subset of metadata from the IEC System.

The next four chapters present the research and results of this thesis. Chapter 2 provides an overview of information retrieval, rule learning methodologies and the WordNet lexical dictionary. Chapter 3 discusses the methodology for implementing this rule learning and rule generalization system. Chapter 4 presents testing and analysis of test results. Chapter 5 concludes the research with conclusions and recommendations for future work.

2. Background

2.1 Introduction

This chapter provides background information useful for establishing a foundation for this research effort. It provides a brief discussion on information retrieval methods, rule learning and lexical reference systems.

2.2 Information Retrieval Methods

Information retrieval (IR) methods include systems for indexing, searching and recalling data, particularly text or other unstructured forms. While there are a number of methods, the three most widely used and well known are the Boolean, probabilistic, and vector methods.

2.2.1 Boolean Method

The Boolean method uses a set of keywords associated with each record within a system. These keywords are the index terms. Users type in one or more of these index terms to retrieve records that match these terms. The Boolean operators are *and*, *or* and *not*. Mixing two or more terms with one or more Boolean operators refines the search, and can reduce or increase the number of records returned. The combination of these terms is a search query, and the Boolean retrieval system returns records based on these queries.

Let X_n represents a term in a query. In the query [(X_1 *and* X_2) *or* (X_3 *and* X_4) *and not* X_5], retrieved records must contain the term pairs X_1 and X_2, or X_3 and X_4, or both. However, none of the records can contain X_5.

While the Boolean method is widely used, it has limitations and disadvantages. One of the primary disadvantages is that many users have no understanding of Boolean logic. This hinders their capability for building effective queries. Furthermore, Boolean logic is quite unyielding in a retrieval system when using *or*-ed only or *and*-ed only terms. The presence of one of the terms X_n in a record in a query (X_1 *or* X_2 *or* X_3 *or* X_4 *or* X_5) returns that record. Conversely, the absence of only one of the terms X_n in a record in the query (X_1 *and* X_2 *and* X_3 *and* X_4 *and* X_5) rejects that record [SLA91].

Even cogent Boolean search string is limited by the order of returned records. Boolean retrieval methods on large information systems can return huge sets of unordered or poorly ordered records. Since it is now much easier to store vast amounts of information, users must have the ability to retrieve desired records efficiently. Finding capable methods of quickly returning the most relevant information to users is a priority for many in computational research arenas.

2.2.2 Probabilistic Method

Marion and Kuhms first presented the probabilistic approach to information retrieval (IR) [MAR60]. The probabilistic approach seeks to the answer to the question [JON98]:

"What is the probability that *this* document is relevant to *this* query?"

The answer to this question begins with an ordered document set from the entire document collection. The problem is a user does not know what this set should look like unless they inspect each document. Therefore, the probabilistic model provides an initial starting point and adjusts relevance through user feedback compiled over several searches [BAZ99]. Estimating a starting point can be computationally inefficient [CRE98].

2.2.3 Vector Method

The Boolean model assumes that all index terms have an equal weight. IR vector-based systems add a numeric weight to each term, expanding the computational possibilities. This improvesof the system through the application of a variety of probabilistic methods. Such systems are known as relevance feedback systems. In a relevance feedback system, the terms in each document have relevance weights. A query combined with a set of documents creates a new and presumably more useful query [ALL95].

Text categorization is the process of assigning term relevance and frequently uses two approaches. Each approach makes use of a bag-of-words representation that looks at documents as bags-of-words without considering word order. Each approach assigns a value to a set of attributes, sometimes called features, based on the function of the respective approach. In both approaches, each distinct word is a feature and the number of times the word occurs within a document determine its value. Since there is no consideration of word order, some information is lost with this representation [JOA97].

One such method is the Term Frequency, Inverse Document Frequency (TFIDF) approach [JOA97]. This method represents each document as a vector with weights based on TFIDF. Documents with similar content have similar vectors. There is a direct correlation with the angle between two vectors and the number of matching terms. The smaller the angle between two document vectors, the more similar the documents.

TFIDF calculates a vector for a user query and compares the user query vector with all the document vectors, returning an ordered list of documents. TFIDF ranks each document vector with respect to the query vector, using the angle between the two for determining the rank. The smallest angle receives the highest rank. While this method provides more effective retrievals, it also substantially increases computational effort [SLA91].

Another method uses one of the many Naïve Bayes classifying algorithms. These classifying algorithms use Bayes’ rule to simplify computations by assuming all term classes are independent. The classifier determines which class or classes the document belongs in. The algorithm assigns documents to one or more classes and sorts them. User queries can then quickly retrieve classified documents.

Other methods explore a variety of document ranking techniques, such as considering passages derived from complete documents [WIL94], or from clusters of paragraphs, or from arbitrarily lengths of long strings of related sentences [HEA93]. In addition, probabilistic methods applied to searches using TFIDF extract better results than with TFIDF alone [JOA97]. However, most of these methods still rely on structured analysis of the documents’ terms and the query, and do not gain knowledge from a user classifying the results.

2.3 Rule Learning

Rule learning is different from IR methods; concepts in the form of rules are stored and used for future searches. A difficult aspect of any sort of machine learning is the presentation of results in human readable form. If-then rules provide one of the most expressive and understandable forms of knowledge representation [MIT97]. Rule learning algorithms vary in the way they search a training data set and in the way they generalize. They also differ in the way they represent class descriptions as well as how they cope with errors and noise in the training data. The sections following this overview introduce strategies and discuss a number of methods of learning rules in this form.

Concept learning systems are differentiated by the complexity of the input and output languages they use. Learning systems that use propositional approaches lay at one extreme of complexity, logical inference systems at the other. The former lends itself to more simplistic representation, using conjunctions (*and*) and disjunctions (*or*) of proposition terms. This simplicity makes such systems suitable for large volumes of data. They represent concepts as collections of examples and counter examples, and thus can exploit the statistical properties of these collections [QUI90]. The latter accepts descriptions of complex, structured entities, and generates classification rules and expresses them in first-order logic.

2.3.1 Rule Learning Strategy

Learning a concept is achieved through one of two methods; simultaneous covering algorithms and sequential covering algorithms [ART99]. The first category includes decision trees, while the second includes direct rule learning algorithms.

2.3.1.1 Simultaneous Covering Algorithms

Decision tree algorithms learn the entire set of disjunctions simultaneously as part of one search through a selected decision tree [MIT97]. These algorithms use a strategy of overfit-and-simplify [ART99]. The algorithms prune results after a search to reduce the generated rule set. ID3 [QUI86] and C4.5 [QUI93] are examples of simultaneous covering algorithms.

ID3 uses a hill-climbing approach to find a locally optimal solution using a greedy technique. This technique branches the decision-tree and selects the feature that provides the highest information gain. This information gain reduce the expected entropy of a decision tree [COH92]. C4.5 is an extension of ID3 that addresses some basic problems in ID3 such as the overfitting of noisy data. [MIT97] defines overfitting.

> "Given a hypothesis space *H*, a hypothesis $h \in H$ is said to **overfit** the training data if there exists some alternative hypothesis $h' \in H$, such that *h* has a smaller error than h' over the training examples, but h' has a smaller error that *h* over the entire distribution of instances."

Additional, C4.5 extensions include the incorporation of numerical attributes and discrete values of a single attribute grouped together to support more complex tests. C4.5 also accepts missing attribute values,and increases accuracy by post-pruning rules after the tree induction.

2.3.1.2 Sequential Covering Algorithms

Sequential Covering Algorithms learn rules one at a time. They compute the subsets of data being covered or the subsets representing the decision class, and choose the best rule among alternative attribute-value pairs [ART99]. This class of algorithms is generally split in to two areas; general-to-specific searches and specific-to-general searches. FOIL [QUI90], FOCL [PAZ97] and FOIDL [MOO95] are examples of sequential covering algorithms.

2.3.1.2.1 General-to-Specific Searches

One rule learning approach learns one rule at a time and organizes the search in the hypothesis space the same way simultaneous covering algorithms do, but follows only the most promising branch in the tree at each step [MIT97 and ART99]. The search begins by considering the most general condition possible, the empty test that matches every instance. Next, add the attribute test that best improves rule performance measured over the training samples. This process is repeated each time adding the attribute test that most improves rule performance. This process continues, greedily adding new attribute tests until the hypothesis reaches an acceptable level of performance. A single descendent is followed at each step whereas simultaneous covering grows a subtree that covers all possible values of the selected attributes.

2.3.1.2.2 Specific-to-General Searches

The converse to the general-to-specific search begins the search process with the most specific rule and gradually generalizes over more positive cases. This search relies on positive examples to compute generalizations of clauses [ESP96].

2.3.2 Rule Learning Methods

The first portion of this research effort deals with the problem of learning rule sets. To familiarize the reader with the basic concepts of rule learning as it applies to IQA, the following example is offered with respect to NAIC's IEC System.

2.3.2.1 IEC Rule Learning Example

Suppose a researcher seeks an image of the left side view of a MIG-21 wheel well. The researcher uses the following key words to search for the desired image:

[left side view MIG 21 Fishbed wheel well]

For the purposes of this example, assume queries are not case sensitive. Records returned include all terms and any combination of terms (terms *or*-ed together). The researcher must manually search for appropriate images. Suppose the following metadata records are returned:

1. [CLOSE RIGHT FRONT UNDERSIDE GROUND PARTIAL VIEW OF A MIG-21 FISHBED BANDT 3046 WITH CZECH MARKINGS DETAILING THE STARBOARD WHEEL WELL REAR SECTION]
2. [CLOSE FRONT UNDERSIDE GROUND PARTIAL VIEW OF A MIG-21 FISHBED BANDT 3046 WITH CZECH MARKINGS DETAILING THE RIGHT (STARBOARD) WHEEL WELL REAR SECTION]
3. [CLOSE INTERIAND PARTIAL VIEW OF A MIG-21 FISHBED BANDT 3046 WITH CZECH MARKINGS DETAILING THE LEFT (PANDT) WHEEL WELL FANDWARD SECTION]
4. [CLOSE LEFT FRONT UNDERSIDE GROUND PARTIAL VIEW OF A MIG-21 FISHBED BANDT 3046 WITH CZECH MARKINGS DETAILING THE PANDT WHEEL WELL REAR SECTION]
5. [CLOSE INTERIAND PARTIAL VIEW OF A MIG-21 FISHBED BANDT 3046 WITH CZECH MARKINGS DETAILING THE LEFT (PANDT) WHEEL WELL REAR SECTION]

Note that the search terms included "MIG 21" with no dash, and all the records contain "MIG-21." This difference exposes a severe limitation of many IR systems.

Without an electronic thesaurus that includes the implication {“MIG 21”$\Rightarrow$“MIG-21”} or some other method of defining them as equal, [MIG 21] as a single search term would not return a record with [MIG-21] in the metadata.

The researcher examines each of these five records and categorizes each as “positive”, “negative” or “non-applicable”. All records returned have a default state of “non-applicable” until changed it to positive or negative. This default ensures that records returned but not categorized do not affect the rule learning algorithm.

If the researcher categorized record 3 as “positive”, and records 1 and 2 as “negative” matches, and does nothing to records 4 and 5. The learning algorithm looks at the search input and the metadata of the records categorized as both good and bad and forms rule sets. The rules represent the knowledge that when “MIG 21” is entered image 3 is preferred over others.

After each query and categorization, the system develops one or more rules that define both good and bad responses for a set of search terms. If the researcher queries the system again with the exact same terms and the database information is unchanged, the results will include the records 3, 4, and 5. The records are also ordered with record 3 first, followed by records 4 and 5.

These rules are stored as disjunctions (*or*) of conjunctions (*and*), in the form:

$$t_1 \wedge t_2 \wedge t_3 \wedge ... \wedge t_{n-2} \wedge t_{n-1} \wedge t_n$$

where *ti* one of the rule terms and $\wedge$ is the conjunction symbol. Rule sets on different lines form the disjunctions (*or*) of conjunctions. Each rule also has a predicate associated with it representing terms included in the metadata

Records 4 and 5 follow record 3 as they contain terms that are still within the search criteria, but have not been categorized. Depending on the rule learning method, records 1 and 2 follow records 4 and 5 being least relevant based upon existing rules, or the returned set of relevant records excludes them.

2.3.2.2 FOIL

Quinlan [QUI93] describes FOIL as "…a learning system that constructs Horn clause programs from examples." It uses a separate-and-conquer approach rather than a divide-and-conquer approach [PAZ92]. FOIL is a non-incremental learner that uses a hill-climbing technique guided by a metric based on information theory [PAZ92]. FOIL inductively generates Horn clauses similar to the way ID3 generates decision trees using attribute-value tests [QUI86]. The difference is FOIL measures information gain and uses it to classify examples that have higher gain.

FOIL has two basic operations, starting a new empty clause, and adding a term to the end of that clause. The second operation repeats until no negative example is covered. The process repeats until the set of clauses cover all positive examples. FOIL finds definitions from relations iteratively using this method.

FOIL includes efficient methods adapted from attribute-value learning systems and develops inexact but useful rules. It also can find recursive definitions, but does not possess the capability to express functions within Horn clauses. FOIL requires training sets that include both positive and negative examples, and cannot form new predicates. Finally, FOIL is based on a short-sighted, greedy algorithm which can be computationally very expensive.

2.3.2.3 First-Order Combined Learner (FOCL)

FOCL is an extension of FOIL that incorporates a variety of background information to expand the class of solvable problems. The background information takes the form of rules and represents domain knowledge. With no background information, FOCL is equivalent to FOIL [PAZ92]. The addition of background information takes advantage of domain knowledge which decreases the explored hypothesis space and increases the accuracy of the learned rules.

This background information is broken down in to three class extensions. The first class provides a method for FOCL to limit the search space. The second extension allows FOCL to use predefined rules outside the FOIL rule constructor. The third extension allows the user to input a partial rule that is possibly incorrect. FOCL initially approximates the predicate of the rule being learned. This particular extension makes FOCL somewhat analogous to an inductive learning system [PAZ93].

2.3.2.4 First-Order Induction of Decision Lists (FOIDL)

FOIDL is another extension of FOIL. FOIDL modifies FOIL by representing background knowledge as a logic program. FOIDL neither uses nor constructs explicit negatives examples but quantifies over-generality by estimating the number of negative examples covered. FOIDL represents a learned program as a first-order decision list. This approach provides a useful representation for problems with specific exceptions to general rules [MOO95].

2.4 Lexical Reference Systems (WordNet)

2.4.1 Background.

WordNet was created at Princeton University in 1985, when a group of psychologists and linguists undertook the development of a lexical database [MIL90]. WordNet is an electronic dictionary that divides words into the categories of nouns, verbs, adjectives, and adverbs. While WordNet has the same information as dictionaries and thesauri, it also has many other features beyond definitions, synonyms, and antonyms.

2.4.2 Terms and Definitions.

The WordNet organization structure consists of semantic relations, which are relationships between meanings [MIL90]. These meanings have 5 categories: synonyms, antonyms, hyponyms/hypernyms, meronyms and morphological relations. The meanings of the first two terms are well understood, but the other three require definitions. Examples are provided to clarify their use within WordNet.

2.4.2.1 Hyponyms/Hypernyms.

Two words, *x* and *y*, are hyponyms if a relationship is expressible as "*An x is a (kind of) y*" [MIL90]. For example, {*beagle*} is a hyponym of {*dog*}, and {*dog*} is a hyponym of {*mammal*}, while {*mammal*} is a hypernym of {*dog*}. Therefore, instead of a lexical relation between word forms as with synonyms and antonyms, hyponymy and hypernymy reference relationships between word meanings.

This relationship is transitive. If the relation holds between a first element and a second and between the second element and a third, the relation also holds between the first and third elements. For example {*beagle*} is a hyponym of {*mammal*}. The relationship is also asymmetric, so {*dog*} is not a hyponym of {*beagle*}. These relationships allow the expression of hyponyms and hypernyms in a hierarchical semantic structure placing a hyponym below its superordinate. A hyponym inherits all the features of its superordinate and adds at least one feature that differentiates it from its superordinate, as well as from other hyponyms of its superordinate [MIL90]. The conventions of hyponyms and hypernyms provide the fundamental organizing principle for nouns in WordNet [MIL90].

2.4.2.2 Meronyms/Holonyms.

Two words, *x* and *y*, are meronyms if a relationship is expressible as "*x has a y*": e.g., {*hand*} is a meronym of {*thumb*}. A holonym is the inverse to this relationship; {*thumb*} is a holonym of {*hand*} [MIL93]. Meronym relations are transitive with some qualifications and asymmetrical [MIL90]. WordNet constructs hierarchies using meronyms, yet this is complex in many instances because a single meronym can have many holonyms.

2.4.2.3 Morphological Relations

The morphology of a word form is an important consideration in the practical application of WordNet. The differences between singular and plural nouns and the tenses of verbs although conceptually simple are difficult for computers. For example, if a person looks up the word *flowers*, WordNet should not respond by saying *flowers* is not in the database whenever *flower* is present. The current implementation of WordNet includes morphological complexities of plural nouns and the tenses of verbs [WOR03].

2.4.2.4 Semantic Components of Nouns

WordNet partitions nouns under a set of semantic primes. Table 2-1 shows this set of primes [MIL93]. These primes are the beginning, or prime semantic component of all the words structured below it. While these sets vary greatly in size, they are not mutually exclusive, meaning some words are included under more than one prime. Words included under more than one prime have more than one sense. A lookup of WordNet online for the word *pen* shows the following [WOR03]:

> “The **noun** "pen" has 5 senses in WordNet.
> 1. **pen** -- (a writing implement with a point from which ink flows)
> 2. **pen** -- (an enclosure for confining livestock)
> 3. playpen, **pen** -- (a portable enclosure in which babies may be left to play)
> 4. penitentiary, **pen** -- (a correctional institution for those convicted of major crimes)
> 5. **pen** -- (female swan)”

WordNet separates words contained within each of these groups in to individual files. These files are relatively shallow in a hierarchical sense. Lexical inheritance systems rarely go more than ten levels deep and most that venture that deep are technical in nature. The prime list builds the foundation for the noun arrangement in WordNet. All nouns fit in to one or more categories (when a word has a dramatically different sense.) This is important when considering semantically related terms, which Section 2.4.4 explores.

Table 2-1: Unique beginners for WordNet Nouns

List of 25 unique beginners for WordNet Nouns	
{*act, action activity*}	{*natural object*}
{*aminal, fauna*}	{*natural phenomenon*}
{*artifact*}	{*person, human being*}
{*attribute, property*}	{*plan, flora*}
{*body, corpus*}	{*possession*}
{*cognition, knowledge*}	{*process*}
{*communication*}	{*quantity, amount*}
{*event, happening*}	{*relation*}
{*feeling, emotion*}	{*shape*}
{*food*}	{*state, condition*}
{*group, collection*}	{*substance*}
{*location, place*}	{*time*}
{*motive*}	

2.4.3 Adjectives and Semantic Roles

The primary function of an adjective is the modification of a noun. WordNet categorizes adjectives as descriptive or relational. Descriptive adjectives express a value of an attribute to a noun [FEL93]. To say *The man is tall* assumes there is an attribute *Height* such that *Height(man) = tall*. Reference-modifying adjectives refer to the temporal status of a noun, such as the *former chief of staff,* or the *occasional drink*. Others are intensifying, such as *mere* or *virtual*.

Adjectives are treated completely different than nouns in WordNet. They have both synonyms and antonyms. Curiously, when two or more adjectives are synonyms it is rare (if ever) that they have the same antonyms. WordNet handles this by using synonym sets, called synsets. Character tags within the synsets discriminate between synonyms and antonyms which allows a computer to find a close match to an adjective, by looking for a synonym of an antonym.

2.4.4 Semantically Related Terms

WordNet can identify semantically related terms, an important ability when generalizing a term or set of terms. Several identification methods exist for syntactically related terms. For verbs and adjectives, WordNet uses synsets to determine like terms. This is useful when matching a query adjective with metadata adjectives since synonyms are relatively equal. WordNet finds similar nouns by looking for the hyponym of the superordinate to the query noun. It also considers any noun pairs with the same superordinate as similar.

Figure 2-1 shows a hierarchy for pen and pencil in WordNet. From the previous information, a search for a "*lead pencil*" could be strongly generalized to a "*pencil*" with a semantic distance of 1, and less strongly to "*slate pencil*" with a semantic distance of 2. There is a semantic distance of 4 between lead pencil and ballpoint (pen). This represents a weaker generalization, but still a valid one since both terms fall under the hierarchical level of writing implements.

If a noun (or adjective) is replaceable with no loss in meaning, then a tight synonymous relationship exists between the two difference terms [BRE99]. The semantic distance between any two terms infers a relationship of some weight. The phrase "*big lead pencil*" is replaceable with "*large lead pencil*" with no change in meaning. These have a tight synonymous relationship. The phrases "*large pencil*" or "*large slate pencil*" replace "*big lead pencil*" with less precision, but weights are computable by using semantic distances of all the different terms.

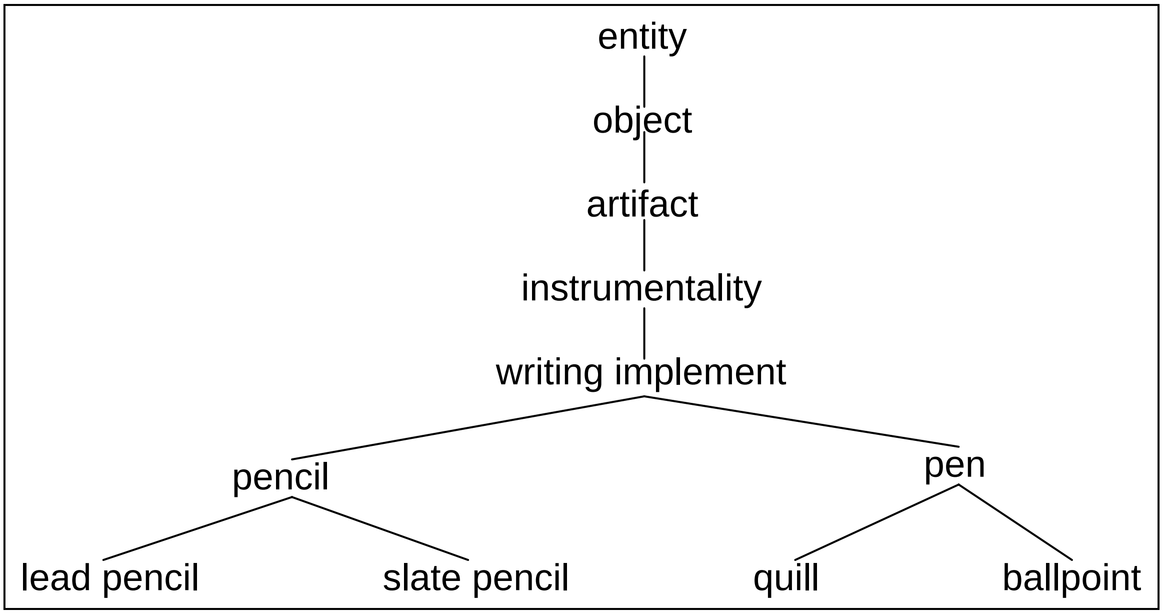


Figure 2-1: WordNet tree expansion of pencil and pen

2.5 Summary

This chapter reviews several topics on IR and rule learning essential for an understanding of IQA. Additionally, WordNet tree expansions, such as the one described in Figure 2-1 provide the model for building a semantic tree in IQA. The concept of semantic trees provides the basis for generalizing terms. Searching this tree for semantically close terms with existing rule associations makes generalizing possible. The next chapter defines and details the IQA methodology.

3. Methodology

3.1 Introduction

Improving methods for searching though a data structure is a widely researched problem. User defined searches usually involve a term or series of terms that is matched against keywords in the data structure. Any match in one or more keywords during a search causes that document to be the returned. Based upon a metric, the system presents ordered results.

Generic database system searches usually return a set of records in no predefined order. As chapter two discussed, several approaches improve this by sorting data via a metric. However, many of these techniques require substantial preprocessing and updating after each data modification.

For web searches, this metric could be the number of page hits a site receives. More recent advances in web searching include counting such things such as back links to the target site. A back link is an instance of one web page containing a hyperlink to the target page. The Google search engine sorts based on the number of back links to a particular web page combined with number of page visits for constructing a page rank [BRI98]. The back link calculation requires crawling the Internet and World Wide Web (WWW) with multiple systems. This method, then, relies on substantial preprocessing of numerous web pages for good results.

This research introduces Intelligent Query Answering (IQA) to develop the relevance metric. IQA techniques offer many benefits over searching databases. One is the system learns user-specific rules. Another is the elimination of information preprocessing. Searches find existing rules, generate pseudo rules, and return matching documents. Document classification builds new rules and reinforces (either positively or negatively) rule weights.

The first portion of this chapter clarifies the specific research objectives. The next section presents the research methodology. The bulk of the chapter discusses the details of data preparation, document search, FOIL and generalization

3.2 Research Objectives

This research develops new techniques that quickly return the most relevant records in database searches, as well as good information on semantically related searches never seen. This research includes the design, implementation, and evaluation of a rule learning and rule generalization system that returns the most relevant records based upon learned and or generalized rules. Specifically, IQA learns rules using a modification of Quinlan's FOIL algorithm and adapt rule weights based upon user classifications and the Winnow algorithm [BLU97]. On subsequent searches, IQA searches a semantic tree for semantically similar rules, and builds pseudo rules for the current search. These pseudo rules provide additional documents and ordering relevance. From the user classified returned documents, IQA learns new rules and reinforces existing rule weights.

3.3 Solution Methodology

The solution is a multi-tiered approach to adequately order records through rule learning and generalization. The solution occurs incrementally, as data flows through the system. The first step prepares the test data and NAIC data for the IQA system. The next tier builds a semantic structure that facilitates rule generalization and rule promotion. The subsequent phases discuss the overall implementation of IQA, including search and user classification, rule learning, and finally rule generalization.

3.4 Data Preparation

Preprocessing is necessary to reduce the amount of work done by the IQA system. The NAIC data source is an image database with corresponding metadata and consists of 3265 declassified records. The data used for the IQA system comes from the comments field (CMMNT) in the NAIC data file.

Two other fields provide contextual clues for the semantic tree building process. The first field is the subject field (SUBJECT). This field identifies the type of object the image represents such as an aircraft or a guided missile. The second field is the subject description (SUBJECT_DES). This field gives the NATO designator of the object, and often includes additional information about image type, such as MIG-21 AFTERBURNER.

Both of these fields are useful to determineing the structure of the semantic tree by providing clues to look up words to look up in WordNet. For example, MIG-21 and FISHBED would not be in WordNet, however, aircraft and fighter are. These clues provide the majority of ideas for developing an appropriate hierarchical structure.

Data preprocessing exacts the data from an ODCB compliant database and saves it to a text file with one field per record. It separates records with line feeds and removes all extraneous characters (punctuation, double spaces and parentheses.) Preprocessing also removes records with exact duplicate data in the CMMNT field, as well as prepositions, conjunctions and determiners. These modifications result in 2617 records and 6652 unique keywords. The resulting words are loaded in an appropriate Java object structure.

3.4.1 Semantic Tree Building

WordNet is the inspiration for building a semantic tree, however, WordNet is not integrated into IQA for two reasons. First, the NAIC data contains many military terms and NATO weapon system designations not in the WordNet database. To correct this deficiency, all non-existing terms would have to be placed in WordNet using the "grinder." The second reason is the complexity of integrating WordNet into IQA along with adding search generalization. For these two reasons, a custom semantic tree with search term generalizer is developed.

WordNet's semantic structure provides a foundation for rule generalization. The concept of hypernyms and hyponyms provides a traversable tree structure for finding semantically similar words and quantify their distance from each other. Hyponyms identify the relationship of two words described as "*an x is a (kind of) y*." In this example, *x* is a hyponym of *y*, while *y* is a hypernym of *x*. Figure 3-1 shows a small semantic tree. Appendix A lists the NAIC data included in its semantic tree.

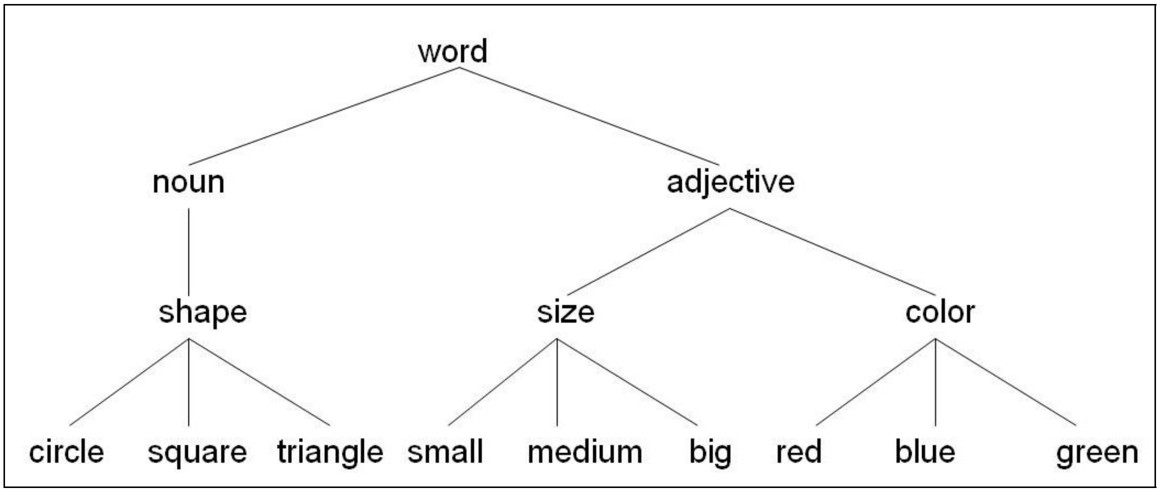


Figure 3-1: Shape Data Semantic Tree

WordNet concepts also assist with finding hypernyms and hyponyms for the NAIC terms as mentioned earlier. When WordNet does not find a military term or NATO designator, then the logical hypernym is used. The level of detail in the semantic tree should reflect the needed granularity to effectively generalize rules.

A hash table represents the IQA semantic tree. IQA builds this hash table from a text file each time IQA runs. This file contains the primary word and pointers to any hyponym(s) associated with it. Each entry also includes a pointer to its hypernym, unless the word is the semantic tree root. The final entry is a Boolean flag used to determine whether the current search iteration has visited this entry (node). This flag is the only data item that changes in the semantic tree during IQA execution after the initial creation of the semantic tree hash table. Finally, IQA assumes that all query terms exist in the semantic tree.

3.4.2 Test Data

A small example set of data facilitates the design, coding and testing of IQA. This test data consists of a class of shapes with attributes of size and color. There are three different shapes, as well as three of each type of attributes. Figure 3-1 shows a visualization of the semantic tree that the test data hash table represents.

3.5 IQA Implementation

IQA includes five major areas; document search, document classification, rule learning, rule generalization, and rule rewrite. The following sections present a detailed description of IQA's implementation.

3.5.1 Document Search

When a user performs a search, the IQA system executes four steps. The first step generates a found rule list, followed by the generation of a pseudo generalized rule list. IQA then finds all documents matching any the search terms and rules on both rule lists, and then and generates a document match list. IQA then assigns a relevance score to each document and returns that ordered document match list for user classification.

3.5.1.1 Found Rule List

IQA stores new and updated rules to a disk file after each search. Rule search terms, rule document terms and a rule weight make up each rule object. The user enters the search terms *st* to find existing rules with matching rule search terms (*rst)*. An exact match occurs when for all terms x_n, $(x_n \in st) \wedge (x_n \in rst) \wedge (|st| = |rst|)$. That is $st \equiv rst$. Search terms can have one or more matching rules.

3.5.1.2 Pseudo Generalized Rule List

The strength of IQA is its capacity for building rules from semantically related terms. The generation of a pseudo-generalized rule list has four steps; finding rules, removing duplicates, generalizing terms and adjusting rule weights.

3.5.1.2.1 Semantically Similar Rule Search

IQA iterates through each search term and searches the semantic tree to find rules that exist within the semantic distance (δ) threshold of the current term. When it finds a rule within the δ threshold, IQA stores the rule to a pseudo generalized rule list. This continues until IQA exhausts all the search terms.

3.5.1.2.2 Duplicate Rule Removal

IQA then removes (prunes) duplicate rule objects in the list by a direct comparison. Duplicate rule objects occur because rule objects with multiple terms have multiple associations with terms in the semantic tree. For example, the rule [RED CIRCLE] $\Rightarrow$ [BIG RED CIRCLE] has two rule search terms [RED] and [CIRCLE]. A search for the terms [BLUE CIRCLE] reveals no rules, so IQA then begins a search of the semantic tree to find semantically close rules. This search returns a rule associated with the term [RED] and one associated with the term [CIRCLE]. In this instance, the rules are the same. IQA returns both rules because the term [BLUE] is a δ of 2 from [RED], and the term [CIRCLE] is exactly matched (δ=0). The total δ of all found terms is less than or equal the δ threshold. There are two identical rules returned by semantic generalization, so IQA deletes one.

3.5.1.2.3 Term Generalization

The generalization step involves a swap of the search term and rule term. Consider the search terms [BLUE CIRCLE] and the list including one rule [RED CIRCLE] ⇒ [BIG RED CIRCLE]. The swap function swaps the semantically similar terms ([BLUE] for [RED]) in the pseudo generalized rule object. The resulting pseudo generalized rule is [BLUE CIRCLE] ⇒ [BIG BLUE CIRCLE].

3.5.1.2.4 Generalize Rule Weight Adjustment

The final step computes the pseudo generalized rule weight. The generalize rule weight (*grw)* is based upon the following equation.

$$grw = orw * (\frac{1}{1+\delta} + \frac{1}{2+rc}) \qquad [3\text{-}1]$$

The *grw* equals the original rule weight (*orw* multiplied by: the sum of the reciprocal of 1 plus the δ and the reciprocal of 2 plus the rule count. The rule count equals the total number of rules returned with the original search terms.

The second half of the equation ensures that the original rule weight will never be greater than a generalized rule. If there are no rules found at the node terms (*rc=0*), but there is a rule found with a δ of 1 (at the parent, δ=1)), then the second half of the equation will equal 1. Note that the δ will always be at least 1. This means the generalized rule weight at the parent is not scaled down if the search terms do not return a rule at the root terms. This is highly desirable since the purpose of the system is to generalize rules whenever possible.

If there are one or more rules returned for the original search terms, the rule weight scales down by a factor that directly depends upon the number of those rules, as well as the δ of the generalized rule found. For a set of search terms that finds no matching rules (*rc=o*), but does finds one rule to generalize that is a δ of 2 away from the original term, the generalize rule weight equals:

$$grw = orw * (\frac{1}{1+2} + \frac{1}{2+0)}) = orw * (\frac{5}{6}) \qquad [3\text{-}2]$$

3.5.1.3 Document Match

During search, IQA compares the disjunction of search terms to each document and adds any document with one or more matching search terms to a document match list. It then compares the rule search terms of rule objects on both rule lists to all documents and adds any document with an exact match to the document match list, exactly like the find rule match in Section 3.5.1.1.

3.5.1.4 Relevance Score

A document's relevance score consists of the number of original search term hits, rules weights of satisfied rules and pseudo generalized rules satisfied, and an additional value of 2 for each rule satisfied. The document relevance score (rs_d) is:

$$rs_d = \frac{n}{sw}(\sum_{i=1}^{n}((sw + rw_i) * rw_i)) + 2rm \qquad [3\text{-}3]$$

In this equation, n is the number of rules satisfied, rs_d is the relevance score, sw is the search weight, rw_i is the rule weight of the rule satisfied, and rm is the number of rule matches.

Rule matches (*rm*) increase the relevance score by 2 for each rule matched. The other portion of the *rs* equation depends on a summation of the *sw* and *rw* for each rule matched. The multiplication of *rw* differentiates ensures $rw < 1$ force a smaller (often times much smaller) overall *rs*. The relevance score increases with the number of matching rules, as well as matching pseudo generalized rules.

IQA prunes the returned document list to reduce the number of documents returned in large rule sets. IQA removes documents with a relevance score of less than $(0.5)st$ where *st* is the number of original search terms. Searches with large numbers of terms can return large numbers of documents even without rules, since the term search uses a disjunction of terms (or-ed.) This pruning prohibits IQA from returning superfluous (ultra-low scoring) documents.

3.5.2 User classification

IQA presents the user with a matching document list sorted by relevance score from highest to lowest. A matching document consists of the search terms, the document terms and the relevance score. The user classifies each returned document as positive (good), negative (bad), or non-applicable (not classified.) Once the user has classified the documents, the rule learning process begins.

3.5.3 Rule Learning Process

The rule learning portion of this research uses the FOIL algorithm [QUI91]. The rule learning process consists of FOIL, the gain function and the Winnow Algorithm.

FOIL learns rule terms based on the gain function. Once rules are learned, the Winnow algorithm updates all rule weights.

3.5.3.1 FOIL

FOIL learns sets of first order rules using a separate-and-conquer approach [PAZ92]. The rule search terms are a disjunction of literals while the rule document terms are a conjunction of literals. Figure 3-2 outlines the FOIL algorithm. It starts with an empty rule and loops through positive and negative examples (separate). For each literal, FOIL calculates the information gain using an entropy function, as discussed in section 3.5.3.2. The literal with the highest gain is added to the antecedent list (conquer). FOIL removes negative examples that do not satisfy the literal and repeats until there are no more negative examples. Once there are no more examples in the negative set, the antecedent list is stored as the next rule.

Let A = {}
Let R = {}
Let P be the current set of uncovered positives
Let N be the set of all negative examples
  Until P is empty do
    Until N is empty do
      For every feature-value pair (literal) $F_i=V_j$, calculate Gain($F_i=V_j$, P, N)
      Pick the literal, L, with the highest gain.
      Add L to A.
      Remove from N examples that do not satisfy L.
    Return the rule: $A = L_1 \wedge L_2 \wedge \ldots \wedge L_n \Rightarrow Positive$
    Add A to R.
    Let N be the set of all negative examples
    Remove from P examples that satisfy A.
Return the rule set: $R = A_1 \vee A_2 \vee \ldots \vee A_n \Rightarrow Positive$

Figure 3-2: FOIL Algorithm

FOIL adds all negative examples back to the original set, and all positive examples that match the new rule are removed from the positive set. If there are still positive examples remaining, IQA creates a new rule with no terms (separate). The process continues until no more positive examples exist. The result is one or more rules that best cover the positive examples.

3.5.3.2 The Gain Function

As FOIL compares each of the search terms (referred to as literals in this section), a gain function chooses which literal to add to a specific rule. *Foil_Gain* is calculated for each literal within a (*conquer*) loop, and the literal with the maximum gain is returned. If there are two or more literals with equal maximum gains, then FOIL uses the first literal for consistency. Equation 3-4 illustrates the Foil_Gain algorithm [MIT97].

$$Foil\_Gain(L,R) = |p| * (\log_2(\frac{|p|}{(|p|+|n|)}) - \log_2(\frac{|P|}{(|P|+|N|)})) \qquad [3\text{-}4]$$

where L represents the current literal candidate for rule R. p represents the subset of examples (documents) in P that satisfy L. n is the subset of examples in N that satisfy L. Foil uses cardinalities of these four terms for computing the gain for a given literaland estimates the utility of adding a new literal based on the numbers of positive and negative examples covered before and after adding the new literal [MIT97].

3.5.3.3 The Winnow Algorithm

Foil adjusts rule weights for all rules using the Winnow algorithm [BLU97]. The rule weight is the basis for the rule strength and is an integral part of the relevance score responsible for returned document order. Each time a user classifies a matching document, FOIL adjusts the associated rule weight. Section 3.5.4.3 discusses the need for adjusting *all* rule weights after each search, regardless of classification or use.

When a user classifies a document as positive, the Winnow algorithm increases that associated rule's weight by multiplying the current rule weight by the positive rule adjustment factor. This reinforces the rule's validity by strengthening the relationship between the rule search terms and the rule document terms. Winnow decreases a negatively classified document's associated rule weight for the contrary reason, weakening the rule strength. Table 3-1 shows the amount Winnow adjusts each rule weight depending on its classification.

Table 3-1: Winnow Adjustment Factor

Classification	Rule Adjustment Factor
Positive (good)	1.5
Negative (bad)	0.5

One category the Winnow algorithm does not address is adjusting rule weights of rules classified as non-applicable or rules not used. This capability is important for two reasons. First, it provides a way of differentiating between two similar rules created during different search iterations. Each rule created starts with an initial rule weight of 1.0. Suppose FOIL creates two similar rules 30 search iterations apart. Without any rule weight adjustment for rules not used, the two rules weights would be the same in a subsequent search that fires both rules. This is undesirable, since a recently learned rule has a higher relevance to the current search criteria. The second reason is for providing a way of identifying a rule that is no longer used.

IQA introduces a specific rule adjustment factor for any rules users do not classify or are not used during a search. Using a rule adjustment factor less than but close to 1.0 for non-classified or unused rules provides a method to subtly reduce the rule weight over time.

Table 3-2: Modified Winnow Adjustment Factor

Classification	Rule Adjustment Factor
Positive (good)	1.5
Negative (bad)	0.5
Not classified	**0.9**

3.5.4 Generalization

Rule generalization is the method of reducing the number of rules in the rule file. IQA uses three separate processes to accomplish this; rule promotion, rule assimilation and rule aging. The rule promotion occurs when a majority of similar rules exist on the same semantic level under a single parent node. Rule assimilation is necessary when FOIL learns a specialized rule and there already exists a more general form of the rule at the parent node on the semantic tree. Rule aging occurs when a rule's weight drops below the usefulness threshold.

The semantic tree is used to determine semantic level, parent node and δ of rules under consideration. Traversing the tree is necessary for determining the need to promote and or assimilate rules. IQA considers only the rule weight when aging rules. The most computationally conservative approach is to iterate through the existing rules and search the semantic tree term by term. This also gives a starting place for the search, which is the first search term of the first rule.

3.5.4.1 Rule Promotion

Rule promotion occurs when IQA combines specific rules into a more general rule. IQA considers rules for promotion when the following criteria is met. First, the rules must be on the same semantic level. Next, the rules must also have the same parent, meaning the rules are separated by a δ of 2. This distinction of δ combined with the same parent is necessary because simply having a δ of 2 could cause an attempted match of a rule with one at its semantic grandparent's level. Since one of the criteria for rule promotion is being on the same semantic level, avoiding this situation conserves computational effort. The final criterion is that the number of similar rules must represent a majority of the total number of siblings under the parent. If IQA considers two rules for promotion and finds that three siblings exist at the semantic level under the parent of consideration, then a majority exists. IQA promotes the rules. This concept is illustrated with two similar rules present under a parent with three siblings:

$$\begin{aligned}&(((st_1 \vee st_2 \vee ... \vee st_n \vee n_y) \Rightarrow (dt_1 \vee dt_2 \vee ... \vee dt_n \vee n_y)) \wedge \\ &((st_1 \vee st_2 \vee ... \vee st_n \vee n_z) \Rightarrow (dt_1 \vee dt_2 \vee ... \vee dt_n \vee n_z))) \Rightarrow \\ &((st_1 \vee st_2 \vee ... \vee st_n \vee p_{y,z}) \Rightarrow (dt_1 \vee dt_2 \vee ... \vee dt_n \vee p_{y,z}))\end{aligned} \quad [3.5]$$

where st_x is a rule search term while dt_x is a rule document term. In each of the two rules, all the terms match with exception of terms n_y and n_z. These terms have the same parent term, denoted by $p_{y,z}$. Since a majority of the siblings have matching rules per equation 3-5, the rules are candidates for promotion. For example, using the shape domain of Figure X, if the two rules [SMALL RED] → [SMALL RED CIRCLE] and [SMALL GREEN] → SMALL GREEN CIRCLE] is generalized to [SMALL COLOR] → [SMALL COLOR CIRCLE]

IQA builds temporary promote rule objects from each rule that fires. One object is built for each rule search term and represents a pointer matching this rule to its search term on the semantic tree. For a rule with two rule search terms, IQA creates two separate promote rule objects. The difference in each of these rule objects is the current node (the rule search term) and the node's parent. This current node provides the semantic location of a rule which is crucial when considering rules for promotion. Table 3-3 shows the promote objects for the rule [SMALL CIRCLE] → [SMALL RED CIRCLE].

Table 3-3: Rule Objects

	Rule Object 1	**Rule Object 1**
Node	SMALL	CIRCLE
Parent	SIZE	SHAPE
Rule	[SMALL CIRCLE] → [SMALL RED CIRCLE]	[SMALL CIRCLE] → [SMALL RED CIRCLE]

Once IQA builds all the rule objects, it compares each rule object to every other. When it finds two rule objects with the same parent, it compares the rule search terms and rule document terms, less the node terms. When a match is found, IQA creates a rule-matching object and stores this information for future use. This continues through each of the promote rule objects.

IQA iterates through rule-matching objects to determine if a majority of matching rules exist for a given parent. If a majority exists, the system marks the first rule for promotion and all subsequent matching rule objects for deletion. The rule-matching object accumulates all matching rule weights for calculating a new rule weight for the promoted rule. If a majority of rules does not exist, then IQA promotes none. If a single promote rule object exists in a matching rule object, then no action is required. Figure 3-3 shows rules before and after a successful rule promotion iteration.

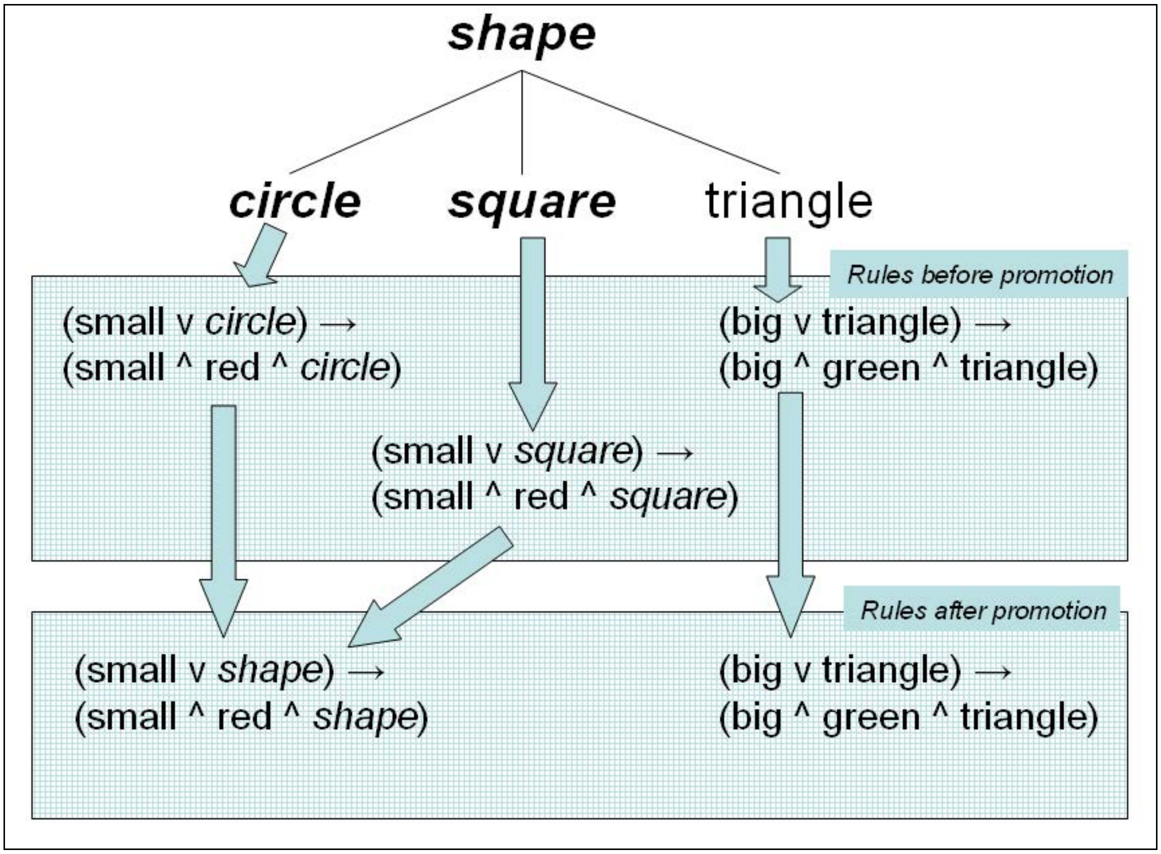


Figure 3-3: Rule Promotion

IQA calculates the promoted rule weight as:

$$prw = \frac{(\sum_{1}^{n} mrw_i)}{m} \times 1.5^n \qquad [3\text{-}6]$$

where *prw* is the promoted rule weight, *n* is the number or matching rules, mrw_i is the matching rule weight, *m* is the total number of siblings under the parent of the matching rules.

This equation takes the sum of all matching rule weights and divides by the total number of node siblings. It multiplies that value by the positive Winnow classification factor of 1.5 taken to the power of the number of matching rules. This applies a positive classification to the promoted rule *n* times. This is desirable and the result is a positive classification for each matching rule. The promoted (generalized) rule weight is greater than the sibling rules in all promotion instances.

3.5.4.2 Rule Assimilation

Rule assimilation occurs when IQA deletes a specialized rule because a more general rule already exists. This process prevents IQA storing learned specialized rules already generalized, thereby diminishing the computational effort of searches.

This process uses the promote rule objects and compares them to each other. For each promote rule, it checks subsequent ones to see if the promote rule node's parent matches the compared promote rule's node. If so, IQA compares the rule search terms and rule document terms to see if they match, less the node terms. If they match, the primary promote rule object is a candidate for assimilation.

During assimilation, IQA marks the candidate for deletion and adjusts the generalized promote rule object rule weight by a factor of 1.5. This adjustment is necessary since the user classified the document returned by this specialized rule as positive. Figure 3-4 illustrates rule assimilation.

3.5.4.3 Rule Aging

Users periodically misclassify returned documents. The possibility also exists that a unique rule never fires again. IQA uses rule aging to delete unnecessary rules from the rules data file. Since many of the IQA functions depend directly on all existing rules, it is computationally desirable to minimize the number of rules.

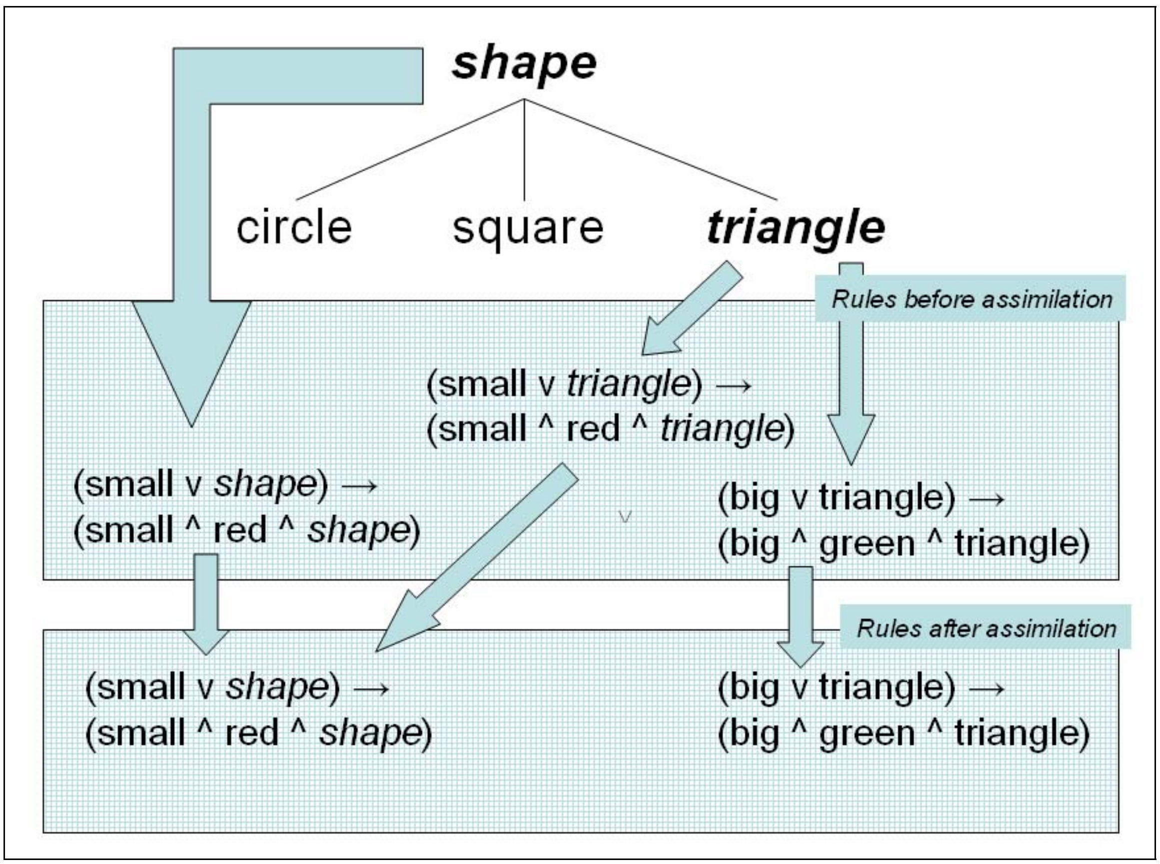


Figure 3-4: Rule Assimilation

The Winnow algorithm provides a way for subtly adjusting the rule weight in cases of rule non-use, or dramatically reducing the rule weight of a rule associated with a negatively classified document. Rule aging flags a promote rule object for deletion if its rule weight drops below a usefulness threshold. This threshold is set at 0.015. If IQA creates a rule that never fires, it takes 40 search iterations to reduce a rule's weight below this threshold. Each time a user classifies a rule as negative, it reduces the number of search iterations before aging by 3. A learned rule classified as negative on subsequent search iterations takes 7 iterations to reduce the rule's weight below the rule aging threshold. Appendix B lists the table that shows the effects of non- or negative classifications on rule weights.

3.5.5 Rule Rewrite

IQA's final step derives updated rules from the promote rule objects. This portion checks each promote rule object to ascertain its deletion status and rule weight. Promote rule objects that have a true deletion status flag or a rule weight below the usefulness threshold are ignored. IQA writes all other promote rule objects to the rules data file. This rule object includes the rule search terms, the rule document terms, and the rule weight. IQA is now ready for the next query.

3.6 Summary

This chapter discusses the design of IQA in detail. It describes the methodology and purpose of each concept, and presents implementations of the most prominent functions in detail. It also illustrates the FOIL algorithm, as well as equations unique and or essential to this application. Examples assist with comprehension of IQA's intent. The next chapter discusses IQA testing, data gathering and results analysis.

4. IQA Evaluation and Results

4.1 Introduction

This chapter describes the techniques used to evaluate IQA. It also presents the results of the main research goal of developing IQA, a rule learning system that generalizes rules across semantically similar words. An analysis follows the empirical results discussing IQA's effectiveness.

FOIL learns rules by greedily adding terms based upon a gain function. An examination of the rule weight follows, ensuring it is initially set properly and updated through future searches per its classification. The next step demonstrates how existing rules can affect the relevance of returned documents and shows an example of how rule weights affect document return order. After IQA builds rules, those rules can be pseudo generalized for current search terms using the semantic tree and existing rules.

The rule generalization function includes rule promotions, rule assimilation and rule aging. Tests demonstrate each of these capabilities and log files verify the expected results. Finally, automated testing compares IQA's ability to return relevant records in a correct sequence based on rule weights, and a positively classified document counter.

4.2 Test Environment

The IQA system uses the Java programming language, JDK version 1.4.1. The software is developed and tested on a computer, with an AMD 2.0 GHz processor with 512 MB of dual-channel RAM.

4.3 Test Data

IQA uses two sets of data for testing. The first set contains one object type (shapes) and two descriptors (colors and sizes), and is referred to as the 'Shape Dataset' from this point forward. IQA uses the 'Shape Dataset' to validate rule learning, the Winnow algorithm, pseudo generalization, and rule generalization (including rule promotion, rule assimilation and rule aging). Table 4-1 shows the terms used for the shape data test set. Combinations of these three term types, one of each type, make up 27 documents that IQA searches. Additional one and two term combinations make up 21 more records, each consisting of at least a shape and a descriptor, bringing the total number of test records to 48. Appendix C includes the 'Shape Dataset.'

Table 4-1: Shape Data Terms

SHAPE	COLOR	SIZE
CIRCLE	RED	SMALL
SQUARE	BLUE	MEDIUM
TRIANGLE	GREEN	BIG

Declassified data from the NAIC IEC system makes up the 'NAIC Dataset.' This data consists of a subset of data from the NAIC IEC system. The NAIC Dataset provides a more realistic test environment for gathering results on document relevance sequence tests.

4.4 FOIL

As discussed in chapter 3, FOIL uses search terms and applicable documents returned in rule building. To show that IQA adds the correct terms to rules from a search, two searches provide gain values of the terms and an evaluation determines if IQA selects the appropriate terms for rules. The FOIL test uses shape data only.

The first test uses the search terms [SMALL TRIANGLE] and returns a set of documents from IQA. There are no learned rules at this point. The tester classifies the document [SMALL RED TRIANGLE] as positive, and all others as non-applicable. IQA assigns gain values for associated terms, shown in Table 4-2.

Table 4-2: Gain Test - [RED TRIANGLE] Search

TERM	GAIN
SMALL	1.000
TRIANGLE	0.585
RED	2.000
Maxterm = RED	2.000
SMALL	1.000
TRIANGLE	0.585
Maxterm = SMALL	1.000
TRIANGLE	1.585
Maxterm = TRIANGLE	1.585

IQA identifies three maximum terms. These terms should make a rule with the search terms of [SMALL TRIANGLE] → [RED SMALL TRIANGLE], since those were the terms that IQA identified as having the maximum gain. Figure 4-1 shows a log output of the log file.

Figure 4-1 shows that the initial search creates one rule, and that rule matches the results shown in Table 4-2. This result is expected since we classified only one document as positive, and that document makes up the rule terms. The possibility exists that two or more terms can have the same highest gain value. When this occurs, FOIL stores the first term with that value as the next rule term. Duplicate gain value become less probable when dealing with less symmetric and equally distributed datasets, such as the NAIC data. Note that the order of document terms in a rule does not affect the performance of IQA.

```
======================================================
>>>>>  NEW RULES <<<<<
======================================================
Total number of new Rules = 1
======================================================
[[SMALL, TRIANGLE], [RED, SMALL, TRIANGLE], 1.0]
******************************************************
======================================================
```

Figure 4-1: IQA Log File Segment- [SMALL TRIANGLE] Search

Next, the tester searches for [RED TRIANGLE] this test uses the current existing data and rule. This time the tester classifies all documents with the terms [RED TRIANGLE] as positive. Table 4-2 shows the gain results. This time IQA identifies two maximum terms, [RED TRIANGLE]. Figure 4-2 shows part of the IQA log file after this test, and the gain values for each.

```
======================================================
Total number of classified results = 24
======================================================
[[RED, TRIANGLE], [RED, TRIANGLE], 1]
[[RED, TRIANGLE], [BIG, RED, TRIANGLE], 1]
[[RED, TRIANGLE], [MEDIUM, RED, TRIANGLE], 1]
[[RED, TRIANGLE], [SMALL, RED, TRIANGLE], 1]
[snip]
======================================================
Total number of Rules = 2
======================================================
[[SMALL, TRIANGLE], [RED, SMALL, TRIANGLE], 0.9]
[[RED, TRIANGLE], [RED, TRIANGLE], 3.375]
******************************************************
[snip]
```

Figure 4-2: IQA Log File Segment - [RED TRIANGLE] Search

4.4.1 Rule Weight Updates

IQA updates all rule weights after each search iteration. IQA assigns a weight of 1.0 to each new rule. Figure 4-1 shows the new rule set after the first search with this value. If a rule returns a document and the user classifies it as positive, then IQA multiplies the current rule weight by 1.5. If a rule returns a document and the user classifies it as negative, then IQA multiplies the current rule weight by 0.5. IQA multiplies the current rule weight by 0.9 under the following conditions:

- the user classifies the document as non-applicable,
- the user does not classify the document, or
- the rule is not used during the current search iteration.

Table 4-3: Gain Test - [SMALL TRIANGLE] Search

TERM	GAIN
RED	2.211
TRIANGLE	2.211
SMALL	0.000
RED	2.211
TRIANGLE	2.211
MEDIUM	0.000
RED	2.211
TRIANGLE	2.211
BIG	0.000
Maxterm = RED	**2.211**
TRIANGLE	4.755
SMALL	0.000
RED	0.000
TRIANGLE	4.755
MEDIUM	0.000
RED	0.000
TRIANGLE	4.755
BIG	0.000
Maxterm = TRIANGLE	**4.755**

If the rule returns multiple documents, then there are multiple classifications and IQA updates its rule weight accordingly. Figure 4-2 shows the rule output from the second search. The rule [SMALL TRIANGLE] → [RED SMALL TRIANGLE] does not fire, and therefore is updated by a factor of 0.9. The rule [RED TRIANGLE] → [RED TRIANGLE] shows a final rule weight of 3.375, counterintuitive to the initial value set at 1.0. This is correct since the user classifies the four documents matching that rule as positive during the same iteration. The initial rule weight is set at 1.0 per the Winnow algorithm. The subsequent rule classifications increase by a factor of 1.5 three times, raising it to 1.5, 2.25, and then finally to 3.375.

4.5 Rule Relevance

IQA uses a relevance weight to sort documents returned for user classification. As discussed in chapter 3, this relevance weight consists of the number of search terms found, the number of rules that match the document and the weights of those rules found. IQA returns documents in relevance weight (also known as relevance score) order, from highest to lowest. This test demonstrates how a rule's weight affects the way IQA sorts returned documents.

The tester now searches for the terms [RED TRIANGLE] in the third search using the rule data from the previous tests. This time the tester classifies the document [BIG RED TRIANGLE] as positive, and all others as non-applicable. The fourth search uses the same search terms. This time the tester classifies only [MEDIUM RED TRIANGLE] as positive and all other documents as non-applicable. Figure 4-3 shows the results for the third and fourth searches. Note that the rule [RED TRIANGLE] → [RED TRIANGLE] has a rule weight that continues to increase. The rule terms match other rule terms while the implied document terms are a subset of other rule implied document terms. This indicates rule validation by positive classification, and therefore the rule weight accordingly.

```
THIRD SEARCH
[snip]
=======================================================
Total number of classified results = 24
=======================================================
[[RED, TRIANGLE], [RED, TRIANGLE], 0]
[[RED, TRIANGLE], [BIG, RED, TRIANGLE], 1]
[[RED, TRIANGLE], [MEDIUM, RED, TRIANGLE], 0]
[[RED, TRIANGLE], [SMALL, RED, TRIANGLE], 0]
[snip]
=======================================================
Total number of new Rules = 3
=======================================================
[[SMALL, TRIANGLE], [RED, SMALL, TRIANGLE], 0.81]
[[RED, TRIANGLE], [RED, TRIANGLE], 3.690562]
[[RED, TRIANGLE], [BIG, RED, TRIANGLE], 1.0]
*******************************************************
[snip]
FOURTH SEARCH
[snip]
=======================================================
Total number of classified results = 24
=======================================================
[[RED, TRIANGLE], [BIG, RED, TRIANGLE], 0]
[[RED, TRIANGLE], [RED, TRIANGLE], 0]
[[RED, TRIANGLE], [MEDIUM, RED, TRIANGLE], 1]
[[RED, TRIANGLE], [SMALL, RED, TRIANGLE], 0]
[snip]
=======================================================
Total number of new Rules = 4
=======================================================
[[SMALL, TRIANGLE], [RED, SMALL, TRIANGLE], 0.729]
[[RED, TRIANGLE], [RED, TRIANGLE], 4.0356293]
[[RED, TRIANGLE], [BIG, RED, TRIANGLE], 0.9]
[[RED, TRIANGLE], [MEDIUM, RED, TRIANGLE], 1.0]
*******************************************************
[snip]
```

Figure 4-3: IQA Log File Segment - [RED TRIANGLE] Search

Each of these four test searches provides a list of unclassified documents sorted in relevant order. Figure 4-4 shows the first two searches, while Figure 4-5 shows the third. Results in italics indicate sort orders of interest, while results in bold indicate which results are subsequently classified as positive. Bold also indicates rules built after classification.

The first log shows the search for [SMALL TRIANGLE]. The top three documents all have a relevance weight of 2.0, so the sort order depends on the order the raw documents were loaded. Since no rules exist, term matching determines the relevance weight. The second and successive searches in this section are for [RED TRIANGLE].

The third search has rule weights associated with them. Since each of the top three records matches the rule, the rule weights are equal and the sort order does not change. IQA builds new rule in the third search as show in Figure 4-5.

FIRST SEARCH	***SECOND SEARCH***
==	==
Total number of unclassified results = 24	Total number of unclassified results = 24
==	==
[[SMALL, TRIANGLE], [SMALL, BLUE, TRIANGLE], 2.0]	***[[RED, TRIANGLE], [RED, TRIANGLE], 2.0]***
[[SMALL, TRIANGLE], [SMALL, TRIANGLE], 2.0]	***[[RED, TRIANGLE], [BIG, RED, TRIANGLE], 2.0]***
[[SMALL, TRIANGLE], [SMALL, RED, TRIANGLE], 2.0]	***[[RED, TRIANGLE], [MEDIUM, RED, TRIANGLE], 2.0]***
[[SMALL, TRIANGLE], [SMALL, GREEN, TRIANGLE], 2.0]	***[[RED, TRIANGLE], [SMALL, RED, TRIANGLE], 2.0]***
[[SMALL, TRIANGLE], [RED, TRIANGLE], 1.0]	[[RED, TRIANGLE], [GREEN, TRIANGLE], 1.0]
[[SMALL, TRIANGLE], [BIG, BLUE, TRIANGLE], 1.0]	[[RED, TRIANGLE], [MEDIUM, TRIANGLE], 1.0]
[[SMALL, TRIANGLE], [GREEN, TRIANGLE], 1.0]	[[RED, TRIANGLE], [MEDIUM, GREEN, TRIANGLE], 1.0]
[[SMALL, TRIANGLE], [SMALL, GREEN, CIRCLE], 1.0]	[[RED, TRIANGLE], [BIG, RED, CIRCLE], 1.0]
[[SMALL, TRIANGLE], [MEDIUM, TRIANGLE], 1.0]	[[RED, TRIANGLE], [BIG, GREEN, TRIANGLE], 1.0]
[[SMALL, TRIANGLE], [SMALL, SQUARE], 1.0]	[[RED, TRIANGLE], [MEDIUM, RED, SQUARE], 1.0]
[snip]	*[snip]*
[[SMALL, TRIANGLE], [SMALL, CIRCLE], 1.0]	[[RED, TRIANGLE], [BIG, BLUE, TRIANGLE], 1.0]
[[SMALL, TRIANGLE], [SMALL, BLUE, SQUARE], 1.0]	[[RED, TRIANGLE], [SMALL, GREEN, TRIANGLE], 1.0]
[[SMALL, TRIANGLE], [BIG, GREEN, TRIANGLE], 1.0]	[[RED, TRIANGLE], [SMALL, RED, SQUARE], 1.0]
[[SMALL, TRIANGLE], [SMALL, RED, SQUARE], 1.0]	[[RED, TRIANGLE], [RED, SQUARE], 1.0]
**	**
[snip]	*[snip]*
==	==
Total number of Rules = 1	Total number of Rules = 2
==	==
[[**SMALL, TRIANGLE], [RED, SMALL, TRIANGLE], 1.0]**	[[SMALL, TRIANGLE], [RED, SMALL, TRIANGLE], 0.9]
**	**[[RED, TRIANGLE], [RED, TRIANGLE], 3.375]**
	**

Figure 4-4: IQA Log File Segment – [SMALL TRIANGLE] and [RED TRIANGLE] Search

Figure 4-6 shows the fourth and fifth searches. The fourth search shows a change in sort order. The fourth search uses the two [RED TRIANGLE] rules to calculate the relevance of returned documents. This increases the relevance weight of document [BIG RED TRIANGLE], and thus moves it to the top of the search order. In the fourth search, a different document is classified as positive, which generates a fourth rule. This fourth rule affects the fifth search by again reordering the top three results.

IQA generates rules on the 'Shape Dataset' in an intuitive way. This is due to the symmetry of the data in both number of terms in each document and the equal term distribution. Document length symmetry and term proportionality reduces testing and debugging complexities. However, testing with non-symmetric and non-proportionate data is highly desired since it is much closer to a live search environment. Appendix C includes the complete log file for these tests.

```
THIRD SEARCH
===================================================
Total number of unclassified results = 24
======================================================
[[RED, TRIANGLE], [RED, TRIANGLE], 20.140625]
[[RED, TRIANGLE], [BIG, RED, TRIANGLE], 20.140625]
[[RED, TRIANGLE], [MEDIUM, RED, TRIANGLE], 20.140625]
[[RED, TRIANGLE], [SMALL, RED, TRIANGLE], 20.140625]
[[RED, TRIANGLE], [GREEN, TRIANGLE], 1.0]
[[RED, TRIANGLE], [MEDIUM, TRIANGLE], 1.0]
[[RED, TRIANGLE], [MEDIUM, GREEN, TRIANGLE], 1.0]
[[RED, TRIANGLE], [BIG, RED, CIRCLE], 1.0]
[[RED, TRIANGLE], [BIG, GREEN, TRIANGLE], 1.0]
[[RED, TRIANGLE], [MEDIUM, RED, SQUARE], 1.0]
[[RED, TRIANGLE], [SMALL, RED, CIRCLE], 1.0]
[[RED, TRIANGLE], [TRIANGLE], 1.0]
[snip]
[[RED, TRIANGLE], [SMALL, RED, SQUARE], 1.0]
[[RED, TRIANGLE], [RED, SQUARE], 1.0]
***************************************************
===================================================
Total number of Rules = 3
===================================================
[[SMALL, TRIANGLE], [RED, SMALL, TRIANGLE], 0.81]
[[RED, TRIANGLE], [RED, TRIANGLE], 3.690562]
[[RED, TRIANGLE], [BIG, RED, TRIANGLE], 1.0]
***************************************************
```

Figure 4-5: IQA Log File Segment – [RED TRIANGLE] Search

4.6 Pseudo Generalization

The term pseudo generalization is the process IQA uses to find semantically similar rules for given search terms. IQA uses those similar terms to build new temporary rules,and returns documents that match those rules. IQA scales the rule weight of those pseudo-generalized rules by a factor based on the number of current rules that match the search terms, and the δ between the original and generalized terms.

Three aspects of pseudo generalization performance are of interest. The first aspect is the creation of pseudo-generalized rules and the effects on document relevance order. The next is the rule weight versus pseudo-generalized rule weight, and how they affect document relevance scores. The final characteristic is the effect of negative document classification of pseudo-generalized rules.

FOURTH SEARCH

```
===================================================
Total number of unclassified results = 24
===================================================
[[RED, TRIANGLE], [BIG, RED, TRIANGLE], 26.001371]
[[RED, TRIANGLE], [RED, TRIANGLE], 23.001371]
[[RED, TRIANGLE], [MEDIUM, RED, TRIANGLE], 23.001371]
[[RED, TRIANGLE], [SMALL, RED, TRIANGLE], 23.001371]
[[RED, TRIANGLE], [GREEN, TRIANGLE], 1.0]
[[RED, TRIANGLE], [MEDIUM, TRIANGLE], 1.0]
[[RED, TRIANGLE], [MEDIUM, GREEN, TRIANGLE], 1.0]
[[RED, TRIANGLE], [BIG, RED, CIRCLE], 1.0]
[[RED, TRIANGLE], [BIG, GREEN, TRIANGLE], 1.0]
[snip]
[[RED, TRIANGLE], [SMALL, TRIANGLE], 1.0]
[[RED, TRIANGLE], [BIG, BLUE, TRIANGLE], 1.0]
[[RED, TRIANGLE], [SMALL, GREEN, TRIANGLE], 1.0]
[[RED, TRIANGLE], [SMALL, RED, SQUARE], 1.0]
[[RED, TRIANGLE], [RED, SQUARE], 1.0]
***************************************************
[snip]
===================================================
Total number of Rules = 4
===================================================
[[SMALL, TRIANGLE], [RED, SMALL, TRIANGLE], 0.729]
[[RED, TRIANGLE], [RED, TRIANGLE], 4.0356293]
[[RED, TRIANGLE], [BIG, RED, TRIANGLE], 0.9]
[[RED, TRIANGLE], [MEDIUM, RED, TRIANGLE], 1.0]
***************************************************
```

FIFTH SEARCH

```
===================================================
Total number of unclassified results = 24
===================================================
[[RED, TRIANGLE], [MEDIUM, RED, TRIANGLE], 29.357563]
[[RED, TRIANGLE], [BIG, RED, TRIANGLE], 26.531805]
[[RED, TRIANGLE], [RED, TRIANGLE], 26.357563]
[[RED, TRIANGLE], [SMALL, RED, TRIANGLE], 26.357563]
[[RED, TRIANGLE], [GREEN, TRIANGLE], 1.0]
[[RED, TRIANGLE], [MEDIUM, TRIANGLE], 1.0]
[[RED, TRIANGLE], [MEDIUM, GREEN, TRIANGLE], 1.0]
[[RED, TRIANGLE], [BIG, RED, CIRCLE], 1.0]
[[RED, TRIANGLE], [BIG, GREEN, TRIANGLE], 1.0]
[[RED, TRIANGLE], [MEDIUM, RED, SQUARE], 1.0]
[[RED, TRIANGLE], [SMALL, RED, CIRCLE], 1.0]
[[RED, TRIANGLE], [TRIANGLE], 1.0]
[[RED, TRIANGLE], [BIG, TRIANGLE], 1.0]
[[RED, TRIANGLE], [BIG, RED, SQUARE], 1.0]
[[RED, TRIANGLE], [SMALL, BLUE, TRIANGLE], 1.0]
[[RED, TRIANGLE], [RED, CIRCLE], 1.0]
[[RED, TRIANGLE], [MEDIUM, RED, CIRCLE], 1.0]
[[RED, TRIANGLE], [BLUE, TRIANGLE], 1.0]
[[RED, TRIANGLE], [MEDIUM, BLUE, TRIANGLE], 1.0]
[[RED, TRIANGLE], [SMALL, TRIANGLE], 1.0]
[[RED, TRIANGLE], [BIG, BLUE, TRIANGLE], 1.0]
[[RED, TRIANGLE], [SMALL, GREEN, TRIANGLE], 1.0]
[[RED, TRIANGLE], [SMALL, RED, SQUARE], 1.0]
[[RED, TRIANGLE], [RED, SQUARE], 1.0]
***************************************************
```

Figure 4-6: IQA Log File Segment – [RED TRIANGLE] Searches

4.6.1 Pseudo-generalize Rule Creation

Rules must exist for pseudo generalization to occur, so the tester searches and classifies a document to create a rule. Then the tester uses semantically similar terms in searching the data ensures IQA generalizes rules within the defined δ. For these evaluations, the δ threshold is 2 for all searches. This ensures that IQA changes at most one search term during pseudo-generalization.

The next search uses the terms [BIG CIRCLE] with no learned rules. The tester classifies the document [BIG GREEN CIRCLE] as positive and the rest of the documents as non-applicable. This generates the rule [BIG CIRCLE] → [GREEN BIG CIRCLE] as shown in Figure 4-7. Note that the document classified as positive is third in the unclassified results list.

```
==================================================
Total number of unclassified results = 24
==================================================
[[BIG, CIRCLE], [SMALL, CIRCLE], 1.0]
[[BIG, CIRCLE], [BIG, GREEN, TRIANGLE], 1.0]
[[BIG, CIRCLE], [BIG, GREEN, SQUARE], 1.0]
[[BIG, CIRCLE], [MEDIUM, CIRCLE], 1.0]
[snip]
==================================================
Total number of classified results = 24
==================================================
[[BIG, CIRCLE], [BIG, RED, CIRCLE], 0]
[[BIG, CIRCLE], [BIG, GREEN, CIRCLE], 1]
[[BIG, CIRCLE], [BIG, CIRCLE], 0]
[snip]
==================================================
Total number of Rules = 1
==================================================
[[BIG, CIRCLE], [GREEN, BIG, CIRCLE], 1.0]
**************************************************
```

Figure 4-7: IQA Log File Segment – Pseudo-generalize Rule Creation for [BIG CIRCLE]

The next search uses the terms [BIG SQUARE] again. Since [SQUARE] has a δ of 2 from [CIRCLE], IQA should create a pseudo generalized rule of [BIG SQUARE] → [GREEN BIG SQUARE]. The existence of this temporary rule forces the document [BIG GREEN SQUARE] to have a higher document relevance weight above all others and thus appear first for classification. Figure 4-8 shows that IQA presents the document [BIG GREEN SQUARE] to the user for classification first and that the relevance weight is indeed greater than other documents with the terms [BIG SQUARE]. This is due to the existence of the semantically similar rule [BIG CIRCLE] → [GREEN BIG CIRCLE]. All returned documents are classified as non-applicable for this test and IQA creates no new rules. IQA also factors the rule weight for [BIG CIRCLE] by 0.9.

```
=================================================
Total number of unclassified results = 24
=================================================
[[BIG, SQUARE], [BIG, GREEN, SQUARE], 3.777778]
[[BIG, SQUARE], [BIG, BLUE, SQUARE], 2.0]
[[BIG, SQUARE], [BIG, RED, SQUARE], 2.0]
[[BIG, SQUARE], [BIG, SQUARE], 2.0] [snip]
=================================================
Total number of Rules = 1
=================================================
[[BIG, CIRCLE], [GREEN, BIG, CIRCLE], 0.9]
*************************************************
```

Figure 4-8: IQA Log File Segment – Pseudo-generalize Rule Search for [BIG SQUARE]

Note the relevance weight for the unclassified document [BIG GREEN SQUARE]. Two search terms match, as well as the matching pseudo-generalized rule. IQA computes the pseudo-generalized rule weight *grw* as $\frac{2}{3}$. The relevance score *rs* is computed as 3.778 as shown below:

$$grw = 1 \times (\frac{1}{1+2} + \frac{1}{2+1}) = \frac{2}{3}, \qquad [4\text{-}1]$$

$$rs = ((2 + \frac{2}{3}) \times \frac{2}{3}) + (2 \times 1) = 1.777778 + 2 = 3.778. \qquad [4\text{-}2]$$

4.6.2 Rules versus Pseudo Generalized Rules

Rule weights have the strongest influence on document's relevance score. If a rule exists for a set of search terms and IQA pseudo generalizes another semantically similar rule, it is plausible that a native rule could have a smaller rule weight than a pseudo-generalized rule, and therefore less of an influence on the overall relevance score. The following example shows just that.

The single rule built in Section 4.6.1 and the 'Shape Dataset' provide the basis for this test. Since a second search yielded no positive or negative classifications, IQA adjusts the rule's weight by a factor of 0.9 as shown in Figure 4-8. The search terms are again [BIG CIRCLE] and the tester classifies only the document [BIG GREEN CIRCLE] as positive. The tester repeats this search and classification to reinforce the learned rule. Figure 4-9 shows the current relevance scores and rule weight.

```
===================================================
Total number of unclassified results = 24
===================================================
[[BIG, CIRCLE], [BIG, GREEN, CIRCLE], 6.5224996]
[[BIG, CIRCLE], [BIG, RED, CIRCLE], 2.0]
[[BIG, CIRCLE], [BIG, BLUE, CIRCLE], 2.0]
[snip]
===================================================
Total number of Rules = 1
===================================================
[[BIG, CIRCLE], [GREEN, BIG, CIRCLE], 2.0249999]
***************************************************
```

Figure 4-9: IQA Log File Segment – Pseudo Generalized Rule Weight

The search terms are now [GREEN SQUARE]. The only rule that exists is [BIG CIRCLE] → [GREEN BIG CIRCLE]. Since [GREEN] is a δ of 2 from [BLUE], and [SQUARE] is also a δ of 2 from [CIRCLE], the combined δ is greater than the δ threshold of 2. IQA should not pseudo-generalize a rule beyond the δ threshold for this test. The combined δ for the current search terms to the existing rule is greater than 2, so there should be no pseudo generalization beyond the δ threshold. The tester classifies the document [BIG GREEN SQUARE] as positive and IQA generates the results shown in Figure 4-10.

The tester repeats the search twice for [BIG CIRCLE] and classifies the document [BIG GREEN CIRCLE] as positive. Figure 4-11 shows the results. The rule for [BIG CIRCLE] is now more than four times as large as the rule for [GREEN SQUARE].

```
==================================================
Total number of unclassified results = 24
==================================================
[[GREEN, SQUARE], [SMALL, GREEN, SQUARE], 2.0]
[[GREEN, SQUARE], [BIG, GREEN, SQUARE], 2.0]
[[GREEN, SQUARE], [GREEN, SQUARE], 2.0]
[[GREEN, SQUARE], [MEDIUM, GREEN, SQUARE], 2.0]
[snip]
==================================================
Total number of classified results = 24
==================================================
[[GREEN, SQUARE], [SMALL, GREEN, SQUARE], 0]
[[GREEN, SQUARE], [BIG, GREEN, SQUARE], 1]
[[GREEN, SQUARE], [GREEN, SQUARE], 0] [snip]
==================================================
Total number of Rules = 2
==================================================
[[BIG, CIRCLE], [GREEN, BIG, CIRCLE], 1.8224999]
[[GREEN, SQUARE], [BIG, GREEN, SQUARE], 1.0]
**************************************************
```

Figure 4-10: IQA Log File Segment – Pseudo Generalized Rule Weight Comparison

```
==================================================
Total number of unclassified results = 24
==================================================
[[BIG, CIRCLE], [BIG, GREEN, CIRCLE], 14.940888]
[[BIG, CIRCLE], [BIG, RED, CIRCLE], 2.0]
[[BIG, CIRCLE], [BIG, CIRCLE], 2.0]
[snip]
==================================================
>>>>>  RULES AFTER PROMOTION <<<<<
==================================================
Total number of Rules = 2
==================================================
[[BIG, CIRCLE], [GREEN, BIG, CIRCLE], 4.100625]
[[GREEN, SQUARE], [BIG, GREEN, SQUARE], 0.81]
**************************************************
```

Figure 4-11: IQA Log File Segment – Rule Weight Reinforcement

The tester now changes the search terms to [GREEN SQUARE] and compares the unclassified results against a search for [BIG SQUARE] without classifying any documents on either search. The search [BIG SQUARE] has no rules associated with it, but it does generalize with the rule for [BIG CIRCLE]. Figure 4-12 shows the difference in relevance weights returned by each respective search. Both searches match one rule and have two matching terms, but IQA generalizes [BIG SQUARE] → [BIG GREEN SQUARE] from [BIG CIRCLE] → [BIG GREEN CIRCLE]. Since [BIG CIRCLE] has a much greater rule weight than [GREEN SQUARE], its relevance score is also higher even when scaled down by the pseudo generalization algorithm.

```
==================================================
Total number of unclassified results = 24
=====================================================
[[GREEN, SQUARE], [BIG, GREEN, SQUARE], 4.2761]
[[GREEN, SQUARE], [SMALL, GREEN, SQUARE], 2.0]
[[GREEN, SQUARE], [GREEN, SQUARE], 2.0]
[[GREEN, SQUARE], [MEDIUM, GREEN, SQUARE], 2.0]
[snip]
==================================================
Total number of unclassified results = 15
==================================================
[[BIG, SQUARE], [BIG, GREEN, SQUARE], 12.505876]
[[BIG, SQUARE], [BIG, RED, SQUARE], 2.0]
[[BIG, SQUARE], [BIG, BLUE, SQUARE], 2.0]
[snip]
```

Figure 4-12: IQA Log File Segment – [GREEN SQUARE] vs [BIG SQUARE] search comparison

4.6.3 Pseudo-Generalized Rules Classified as Negative

IQA learns rules through the standard FOIL algorithm as described in Section 4.4. However, IQA derives pseudo-generalized rules from the process described in Section 4.6.1. If the user classifies all documents returned by pseudo-generalized rules as negative or non-applicable, IQA ignores them.

4.7 Rule Generalization

A pseudo-generalized rule becomes a generalize rule when one or more of the documents IQA returns are classified as positive. This happens concurrently while IQA is using FOIL to learn other rules based upon document classification. This section demonstrates rule promotion and rule absorption functions.

4.7.1 Rule Promotion

Rule promotion refers to taking two or more rules and generalizing one of the terms to a parent term. Figure 4-13 shows an example of this. Rule promotion is desirable to keep the number of rules minimized.

BEFORE RULE PROMOTION
[GREEN TRIANGLE] → [SMALL GREEN TRIANGLE] [BLUE TRIANGLE] → [SMALL BLUE TRIANGLE]
AFTER RULE PROMOTION
[COLOR TRIANGLE]→[SMALL COLOR TRIANGLE]

Figure 4-13: Rule Promotion

This test starts with no rules learned and a search for [GREEN TRIANGLE]. The tester classifies the document [SMALL GREEN TRIANGLE] as positive. The next search is for [BLUE TRIANGLE] and tester classifies the document [SMALL BLUE TRIANGLE] as positive. Figure 4-14 shows the log file results for this test. Note that in the second search, IQA generalizes the [GREEN TRIANGLE] rule for the [BLUE TRIANGLE] search and therefore returns a higher relevance score. Once the FOIL algorithm is complete, the rule promotion function goes through all rules, comparing them with the semantic tree searching for candidates for promotion. In this instance the terms [BLUE] and [GREEN] are both semantic siblings of the term [COLOR]. Since two of the three siblings have a similar rule, IQA promotes (generalizes) them.

IQA can also promote a set of generalized rules. Consider a search for [RED SQUARE] that positively classifies [SMALL RED SQUARE], and then search again for [GREEN SQUARE] and positively classify [SMALL GREEN SQUARE]. In this instance, IQA creates the rule set shown in Figure 4-15. Note that IQA promotes these two rules to the more general rule of [COLOR SQUARE] → [SMALL COLOR SQUARE].

```
=====================================================
Total number of unclassified results = 24
=====================================================
[[GREEN, TRIANGLE], [BIG, GREEN, TRIANGLE], 2.0]
[[GREEN, TRIANGLE], [GREEN, TRIANGLE], 2.0]
[[GREEN, TRIANGLE], [MEDIUM, GREEN, TRIANGLE], 2.0]
[[GREEN,  TRIANGLE],  [SMALL,  GREEN,  TRIANGLE],  2.0]
[snip]
=====================================================
Total number of new Rules = 1
=====================================================
[[GREEN, TRIANGLE], [SMALL, GREEN, TRIANGLE], 1.0]
*****************************************************
{snip to next search]
=====================================================
Total number of unclassified results = 24
=====================================================
[[BLUE, TRIANGLE], [SMALL, BLUE, TRIANGLE], 3.777778]
[[BLUE, TRIANGLE], [BIG, BLUE, TRIANGLE], 2.0]
[[BLUE, TRIANGLE], [BLUE, TRIANGLE], 2.0]
[[BLUE, TRIANGLE], [MEDIUM, BLUE, TRIANGLE], 2.0]
=====================================================
>>>>>  RULES BEFORE PROMOTION <<<<<
=====================================================
Total number of Rules = 2
=====================================================
[[GREEN, TRIANGLE], [SMALL, GREEN, TRIANGLE], 0.9]
[[BLUE, TRIANGLE], [SMALL, BLUE, TRIANGLE], 1.0]
*****************************************************
=====================================================
>>>>>  RULES AFTER PROMOTION <<<<<
=====================================================
Total number of Rules = 1
=====================================================
[[COLOR, TRIANGLE], [SMALL, COLOR, TRIANGLE], 1.425]
*****************************************************
```

Figure 4-14: IQA Log File Segment – Rule Promotion

However, now two rules exist that are candidates for promotion again. IQA takes care of this situation on the next subsequent search. It is computationally more efficient in systems with larger rule sets to perform this check one time after document classification, rather than iteratively searching through rule sets. However, any subsequent search would yield the results shown in Figure 4-16. Note that the terms [TRIANGLE] and [SQUARE] have the same semantic parent of [SHAPE] and are therefore suitable for promotion (generalization), and IQA does promote them in the subsequent search.

```
=====================================================
>>>>>  RULES BEFORE PROMOTION <<<<<
=====================================================
Total number of Rules = 3
=====================================================
[[COLOR, TRIANGLE], [SMALL, COLOR, TRIANGLE], 1.1542499]
[[RED, SQUARE], [SMALL, RED, SQUARE], 0.9]
[[GREEN, SQUARE], [SMALL, GREEN, SQUARE], 1.0]
*****************************************************
=====================================================
>>>>>  RULES AFTER PROMOTION <<<<<
=====================================================
Total number of Rules = 2
=====================================================
[[COLOR, TRIANGLE], [SMALL, COLOR, TRIANGLE], 1.1542499]
[[COLOR, SQUARE], [SMALL, COLOR, SQUARE], 1.425]
*****************************************************
```

Figure 4-15: IQA Log File Segment – Generalize Rule Promotion Before

```
=====================================================
>>>>>  RULES BEFORE PROMOTION <<<<<
=====================================================
Total number of Rules = 2
=====================================================
[[COLOR, TRIANGLE], [SMALL, COLOR, TRIANGLE], 1.0388249]
[[COLOR, SQUARE], [SMALL, COLOR, SQUARE], 1.2824999]
*****************************************************
=====================================================
>>>>>  RULES AFTER PROMOTION <<<<<
=====================================================
Total number of Rules = 1
=====================================================
[[COLOR, SHAPE], [SMALL, COLOR, SHAPE], 1.7409936]
*****************************************************
```

Figure 4-16: IQA Log File Segment – Generalize Rule Promotion After

4.7.2 Rule Assimilation

Rule assimilation is necessary when IQA builds a more specialized rule using FOIL, but a more general rule already exists. Consider the state of the database in Figure 4-13. A search for [RED TRIANGLE] yielded a pseudo-generalized rule from the semantic parent of the term [RED]. However, classifying [SMALL RED TRIANGLE] as positive creates the rule [RED TRIANGLE] → [SMALL RED TRIANGLE] as shown in

Figure 4-17. The relevance score for [SMALL RED TRIANGLE] indicates that IQA creates a pseudo-generalized rule during the search.

Rule assimilation compares this rule to rules at each of the parent terms to see if there is one suitable for absorption. The process of assimilation deletes the sibling rule, and increments the rule weight of the parent rule per the promotion rules. The creation of this sibling rule is akin to a positive classification of the parent rule. In a sense, rule assimilation is a form of rule promotion and therefore occurs in the same manner that rule promotion does.

```
======================================================
Total number of unclassified results = 24
======================================================
[[RED, TRIANGLE], [SMALL, RED, TRIANGLE], 5.7851562]
[[RED, TRIANGLE], [MEDIUM, RED, TRIANGLE], 2.0]
[[RED, TRIANGLE], [BIG, RED, TRIANGLE], 2.0]
[snip]
======================================================
>>>>>  RULES BEFORE PROMOTION <<<<<
======================================================
Total number of Rules = 2
======================================================
[[COLOR, TRIANGLE], [SMALL, COLOR, TRIANGLE], 1.2824999]
[[RED, TRIANGLE], [SMALL, RED, TRIANGLE], 1.0]
******************************************************
======================================================
>>>>>  RULES AFTER PROMOTION <<<<<
======================================================
Total number of Rules = 1
======================================================
[[COLOR, TRIANGLE], [SMALL, COLOR, TRIANGLE], 2.1374998]
******************************************************
```

Figure 4-17: IQA Log File Segment – Rule Assimilation

4.8 Rule Aging

Rule aging is necessary to remove rules that have lost their usefulness. These rules have either gone unused for an extended period, repeatedly been classified as negative, or some combination of both. This provides an automatic way to remove useless rules. The rule-aging limit value is set at 0.015 per the constraints identified in

Section 3.5.4.3. This ensures a rule will age after 40 searches if not classified, 7 search is classified as negative, and somewhere in between if a combination of the two.

Consider Figure 4-7. The tester searches again for [BIG CIRCLE], yet this time classifies [BIG RED CIRCLE] as positive and [BIG GREEN CIRCLE] as negative. This cycle repeats 6 times until IQA deletes the original rule as shown in Figure 4-18. Once the rule weight of any rule has dropped below the usefulness level, IQA deletes the rule.

```
======================================================
Total number of Rules = 2
======================================================
[[BIG, CIRCLE], [GREEN, BIG, CIRCLE], 0.5]
[[BIG, CIRCLE], [RED, BIG, CIRCLE], 1.0]
******************************************************
======================================================
Total number of Rules = 2
======================================================
[[BIG, CIRCLE], [GREEN, BIG, CIRCLE], 0.25]
[[BIG, CIRCLE], [RED, BIG, CIRCLE], 1.5]
******************************************************
 [snip cycles 2-5]
======================================================
>>>>>  RULES BEFORE PROMOTION <<<<<
======================================================
Total number of Rules = 2
======================================================
[[BIG, CIRCLE], [GREEN, BIG, CIRCLE], 0.0078125]
[[BIG, CIRCLE], [RED, BIG, CIRCLE], 11.390625]
******************************************************
======================================================
>>>>>  RULES AFTER PROMOTION <<<<<
======================================================
Total number of Rules = 1
======================================================
[[BIG, CIRCLE], [RED, BIG, CIRCLE], 11.390625]
******************************************************
```

Figure 4-18: IQA Log File Segment – Rule Aging

4.9 IQA versus Document Count Testing

The following automated tests use both the ‘Shape Dataset’ and the ‘NAIC Dataset’ to quantify the capabilities of IQA. These tests provide a comparison of IQA’s ability for ordering documents correctly based upon rules learned. IQA also stores a raw document counter for documents positively classified. The raw document counter keeps track of a value for each document and represents how users classify a document over a period of searches. A positive classification causes the document count to increment by 1, while a negative classification causes a decrement of 1. Non-applicable classifications have no effect. This raw document counter and the relevance weight are the instrumental tools of this test.

IQA collects the order in which it returns documents in two ways. IQA searches for documents and sorts first by relevance weight and then by document count. IQA automatically classifies each document based on two sets of probabilities. The first probability set consists of terms that force IQA to classify a document as positive. The second set consists of terms that force IQA to classify a document as negative. IQA classifies the remaining documents as non-applicable.

IQA resorts the documents by classification, and compares the classified result order to the original order presented. The difference in those orders is stored as a correctness percentage. The plot and analysis of this data provides the running means (averages) of correctness for both types of documents sorts, and allows for a direct comparison of the two approaches.

4.9.1 Probability Sets

Probability sets are only used during automated testing. The probability sets consist of terms, probabilities, and Boolean values that indicate the document classification. These probability sets are referred to as "roulette wheels." There are three wheels used for each test, and they are stored in text files for ease of automation. Figure 4-19 shows an example of each file that represents a separate roulette wheel.

Search Terms	Positive Terms	Negative Terms
2,RED,TRIANGLE,0.50,TRUE 2,SMALL,TRIANGLE,0.80,TRUE 2,MEDIUM,TRIANGLE,1.00,TRUE	1,TRIANGLE,0.60,TRUE 1,RED,1.00,TRUE	1,SQUARE,0.50,FALSE 1,CIRCLE,1.00,FALSE

Figure 4-19: Probability Set Search and Classification Wheel Example

Each line in a file is a record. The record starts with a number that identifies the number of terms that follow. The terms are next, followed by a probability. The probability goes from any value greater than the probability of the previous record (or 0 if it is the first record) to this probability. The final value in a record is the Boolean value that determines the document classification. A value of [TRUE] results in a positive classification, while [FALSE] results in a negative. Negative classifications have priority over positive ones.

The roulette wheels are used in randomly selecting a term or set of terms for an action. The concept of "spinning" a roulette wheel refers to generating a probability used in an IQA search. IQA spins each wheel once for a single automated search. In the example in Figure 4-19, the search terms [RED TRIANGLE] will return any document containing the term [RED]. There is a 20% probability that IQA will classify the document [BIG RED CIRCLE] as positive, so we say that this probability can represent human error in an automated test.

We also introduce the concept of "binding", referring to the number of terms in each positive and negative file. In Figure 4-19, there are four different search terms. Two of those terms are positively bound by one term, while two other terms are negatively bound. If we search for documents with the term [TRIANGLE], then logical we do not want any documents with [CIRCLE] or [SQUARE] classified as positive.

4.9.2 Automated Testing

A tester manually entered search terms and classifications for previous tests, but now automated testing gathers data for analysis. IQA uses a number of Boolean switches to transition between manual and automated testing. IQA also has the ability to repeat a test multiple times with different roulette wheel spins.

Each test begins with no rules and two sets of roulette wheels. The two roulette wheels represent a series of possible searches that encourage rule generalization as testing goes back and forth between the two sets. The only difference between the two sets is one term is swapped with another that is a δ of 2 away. For example, if the term [SQUARE] is replaced with [TRIANGLE], then the term [TRIANGLE] is also replaced with [SQUARE]. This swap assists in inducing generalization, and gives a more realistic scenario for testing.

4.9.3 'Shape Dataset' Tests

The tester conducts both two-term and three-term tests with the 'Shape Dataset.' Each term test set uses the same roulette wheels, less the one changed term to induce generalization. The two-term tests bind 1 term for positive classification, while the three-term tests bind 1 and then 2 terms for positive classification. Each test begins with a randomized set of metadata.

4.9.3.1 2-Term-1 'Shape Dataset' Search

The first test uses the roulette wheels in Figure 4-19 with a negative wheel spin. IQA runs for 50 iterations, with a term swap of [SQUARE] and [TRIANGLE] every 5 iterations. The metadata is randomized and the test is repeated 4 times. Figure 4-20 shows the results with a 95% confidence interval.

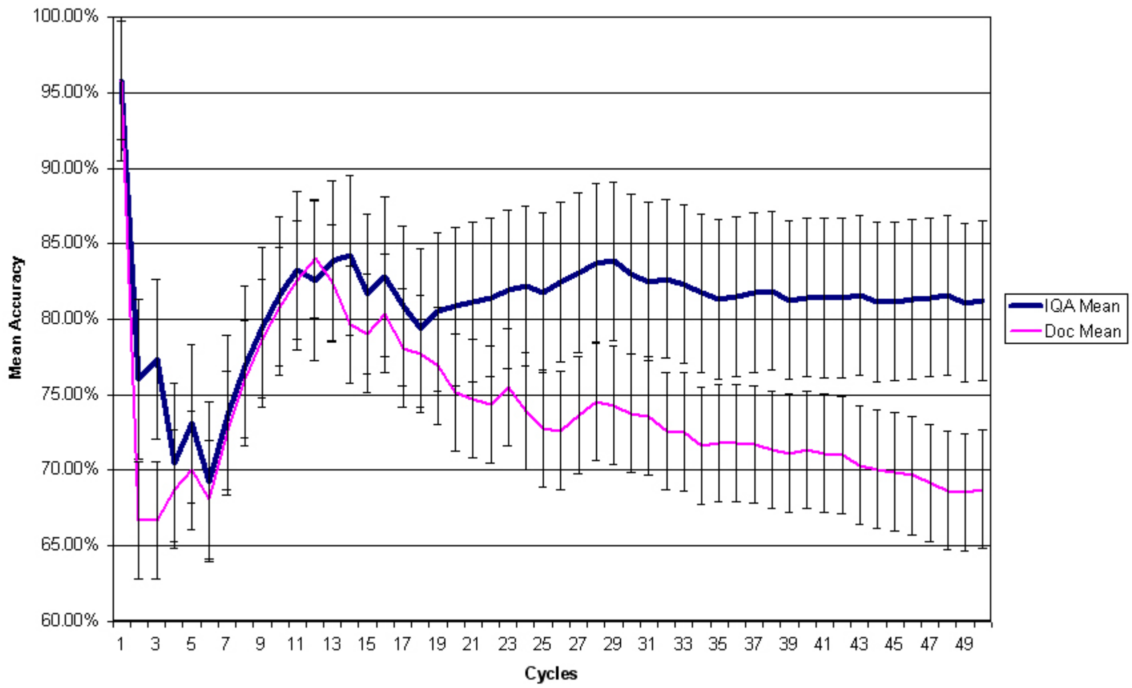


Figure 4-20: Shape Test, 2-Term

The Figure shows the IQA mean accuracy ahead of the document mean for almost the entire test. As IQA generalizes every 5^{th} iteration, its accuracy continues. At iteration 50 IQA has a 12.59% better accuracy rate than the raw document count. Changing terms every five iterations causes the raw document count to stabilize near 67%. This test learns six rules and generalize each rule by one term.

Table 4-4 shows the results from five tests. Raw data shows that IQA performs 5.28% more accurately on average, and outperforms the raw document count in four of five tests. However, the confidence interval shows them to be statistically the same, and only showing a minute increase at a confidence level of 74%.

Table 4-4: Shape Test 2-Term-1

Shape Test	IQA Mean	Doc Mean	Rules Learned	Term Generalized
2 Term 1	71.15%	65.11%	6.00	6.00
	81.23%	68.74%	6.00	6.00
	75.71%	75.79%	4.00	4.00
	82.01%	74.24%	4.00	4.00
	68.41%	68.21%	8.00	7.00
Mean	**75.70%**	**70.42%**	**5.60**	**5.40**
StDev	**6.01%**	**4.45%**	**1.67**	**1.34**
Confidence Level	**IQA Confidence Interval Width**	**Doc Confidence Interval Width**	**Mean Diff**	**Overlap (-)**
95.00%	5.27%	3.90%	5.29%	-3.88%
90.00%	4.42%	3.28%	5.29%	-2.41%
80.00%	3.44%	2.55%	5.29%	-0.71%
75.00%	3.09%	2.29%	5.29%	-0.10%
74.00%	**3.03%**	**2.24%**	5.29%	**0.02%**

4.9.3.2 3-Term-*X* 'Shape Dataset' Searches

The next test uses the roulette wheels in Figure 4-21. The same conditions apply for these tests as they did the two-term tests. The first of these uses one term to classify documents as positive.

Search Terms	Positive Terms	Negative Terms
3,BIG,BLUE,TRIANGLE,0.50,TRUE 3,MEDIUM,BLUE,TRIANGLE,1.00,TRUE	1,TRIANGLE,0.60,TRUE 1,BLUE,1.00,TRUE	1,SQUARE,0.40,FALSE 1,CIRCLE,0.80,FALSE 1,SMALL,1.00,FALSE

Figure 4-21: Probability Set Search and Classification Wheel, Three Term Shape Search

Figure 4-22 shows one of the three-term test results with a 95% confidence interval. Both IQA and the document count outperformed the two-term tests. However, IQA learns rules quickly and generalizes them effectively. The document count accuracy drops considerably at the first generalization point, and does so again at the second. IQA has an accuracy rate 8.18% greater than the raw document count at 50 iterations. This test learns four rules, and all four rules have one generalized term.

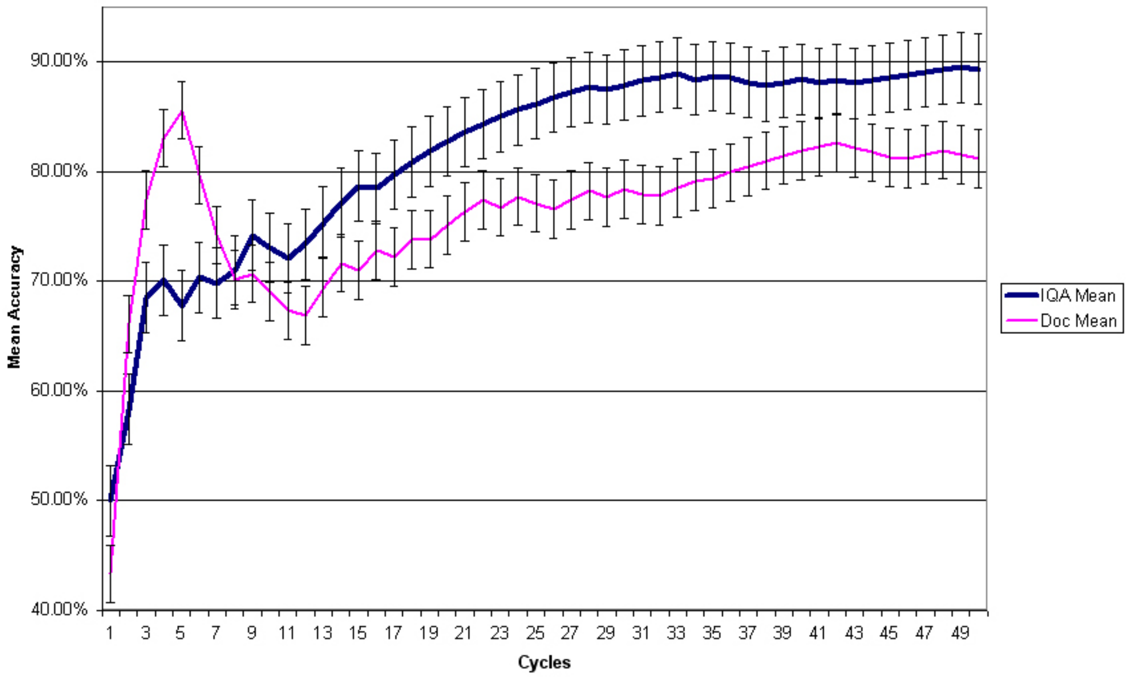


Figure 4-22: Shape Test, 3-Term-1

Table 4-5 shows the results for all 5 tests. IQA's combined results are somewhat better than the raw document count in the previous test. The confidence interval shows a 95% confidence level that IQA will outperform the raw document count by just over 3%, and is slightly better at a 99% confidence level. Coincidentally, IQA learns the identical number of rules and generalizes the same number of term as it did in the 2-Term-1 test.

The next test is also a three-term test, using the classification wheel terms in the previous three-term tests. This time an additional term placed in the positive classification wheel to increase the positive binding. The standard 50 iterations are run with a term swap every five iterations.

Table 4-5: Shape Test 3-Term-1

Shape Test	IQA Mean	Doc Mean	Rules Learned	Term Generalized
3 Term 1	83.87%	75.31%	6.00	6.00
	82.98%	73.30%	6.00	6.00
	89.36%	81.19%	4.00	4.00
	83.20%	76.31%	4.00	4.00
	90.50%	78.36%	8.00	7.00
Mean	**85.98%**	**76.89%**	**5.60**	**5.40**
StDev	**3.64%**	**3.01%**	**1.67**	**1.34**
Confidence Level	**IQA Confidence Interval Width**	**Doc Confidence Interval Width**	**Mean Diff**	**Overlap(-)**
99.00%	**4.19%**	**3.47%**	9.09%	**1.42%**
95.00%	**3.19%**	**2.64%**	9.09%	**3.26%**
90.00%	**2.68%**	**2.22%**	9.09%	**4.19%**
75.00%	**1.87%**	**1.55%**	9.09%	**5.67%**
50.00%	**1.10%**	**0.91%**	9.09%	**7.08%**

The results in Figure 4-23 show IQA overtakes the raw document at 10 iterations (at the first generalization switch) and continues to outperform throughout the test. It finishes 7.89% more accurate than the document count. Generalization switches cause an oscillation in raw document count accuracies early on.

Table 4-6 shows the combined results from all five tests. Even though IQA outperforms the document count again on four out of five tests, the confidence interval check shows these are statistically equivalent. There is only a 24% confidence level that IQA will outperform the raw document count in any way. This time IQA learns fewer rules but still performs almost as accurately as in the 3-Term-1 test. However, the standard deviation for both the methods is much higher. Metadata randomization is the most likely cause, combined with the structure of the probability wheels.

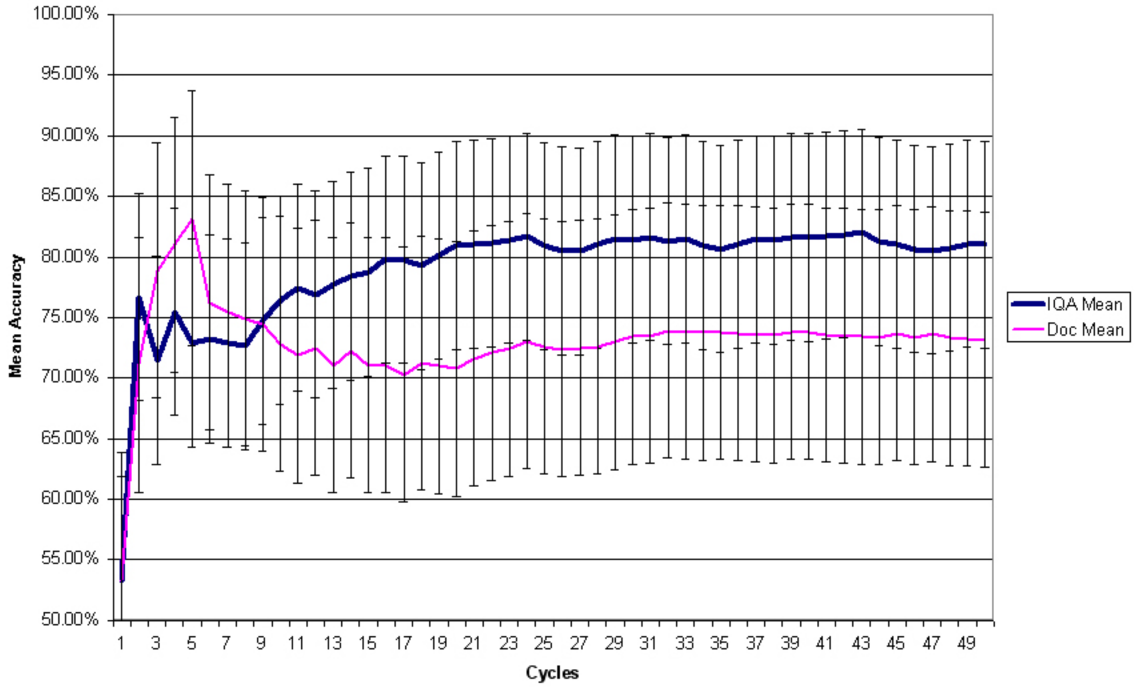


Figure 4-23: Shape Test, 3-Term-2 with stronger binding

Table 4-6: Shape Test 3-Term-2

Shape Test	**IQA Mean**	**Doc Mean**	**Rules Learned**	**Term Generalized**
3 Term 2	93.91%	94.16%	2.00	2.00
	81.04%	73.15%	4.00	4.00
	93.07%	92.53%	3.00	4.00
	72.58%	71.50%	3.00	3.00
	75.97%	70.02%	3.00	3.00
Mean	**83.31%**	**80.27%**	**3.00**	**3.20**
StDev	**9.77%**	**12.00%**	**0.71**	**0.84**
Confidence Level	**IQA Confidence Interval Width**	**Doc Confidence Interval Width**	**Mean Diff**	**Overlap**
95.00%	8.56%	10.52%	3.04%	-16.04%
90.00%	7.19%	8.83%	3.04%	-12.97%
80.00%	5.60%	6.88%	3.04%	-9.43%
75.00%	5.03%	6.17%	3.04%	-8.15%
24.00%	**1.33%**	**1.64%**	3.04%	**0.07%**

4.9.4 'NAIC Dataset' Tests

This data set uses with a 4-Term-1 test and a 4-Term-3 test. The tester performs these identically to the 'Shape Dataset' tests with on exception. Generalization switches occur every 10 iterations and the tests go for 100 iterations.

4.9.4.1 4-Term-*X* 'NAIC Dataset' Searches

We select the roulette wheels as shown in Figure 4-24 for the first test. This test uses the negative wheel spin with four search terms and one bound positive term.

Search Terms	Positive Terms	Negative Terms
4,CLOSE,FRONT,GROUND,MIG-21,0.50,TRUE 4,MEDIUM,DISTANT,INFLIGHT,MIG-21,0.80,TRUE 4,CLOSE,FRONT,PARTIAL,MIG-21,1.00,TRUE	1,MIG-21,0.50,TRUE 1,FRONT,0.80,TRUE 1,CLOSE,1.00,TRUE	1,MIRAGE_2000,0.50,FALSE 1,MIG-21BIS,1.00,FALSE

Figure 4-24: Probability Set Search and Classification Wheel, Four Term NAIC Search

Figure 4-25 shows a 4-Term-1 search test. Both IQA and the raw document count start very accurate. This is attributed to the random way the metadata is sorted. Some sorts present documents in a distribution that closely resembles the way the roulette wheels provide search terms. The document count oscillates with each generalization switch. However, the IQA data becomes less accurate after the 29^{th} iteration. IQA could be over fitting the NAIC data by learning too many rules early on. Both counts stabilize after 70 iterations, but the raw document count ends slightly more accurate than IQA (1.58%.)

Table 4-7 shows the results of all 5 tests. The raw document count outperformed IQA in each of the fives tests. A 95% confidence interval check shows these tests are statistically equivalent. Much of IQA's poor performance can be attributed to extreme over fitting. Only three different sets of search terms are on each roulette wheel. For six possible searches, IQA learned more than 3 rules per test run on average.

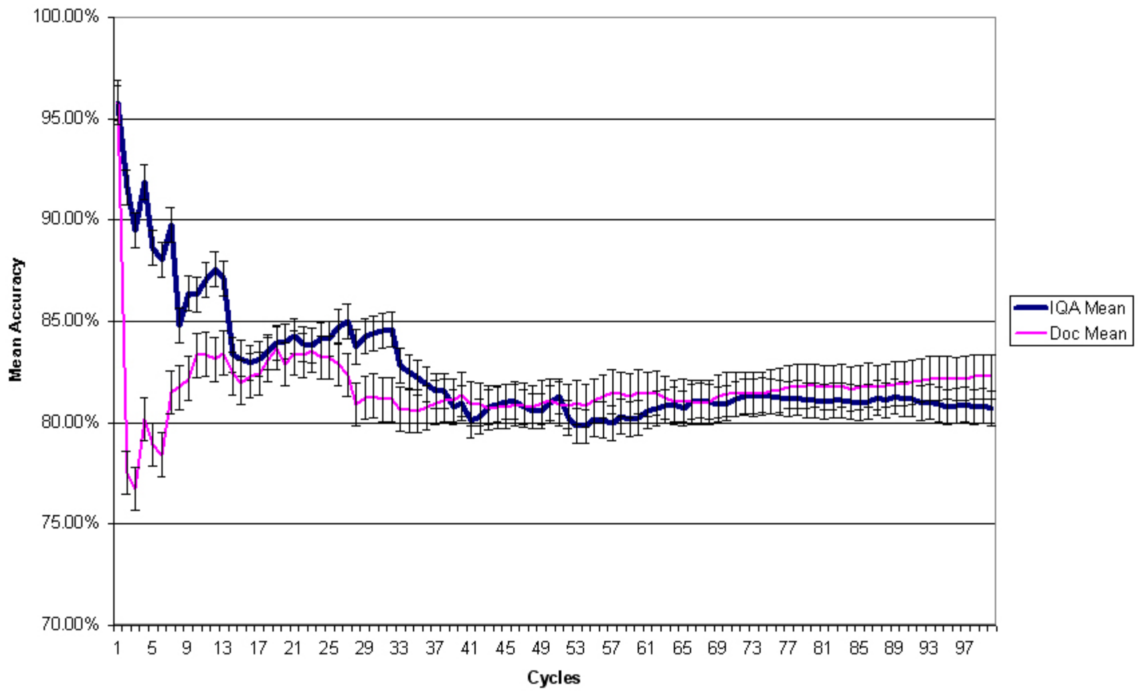


Figure 4-25: 'NAIC Dataset', 4-Term-1

Table 4-7: NAIC Test 4-Term-1

NAIC Test	IQA Mean	Doc Mean	Rules Learned	Term Generalized
4 Term 1	79.16%	80.88%	38.00	1.00
	79.62%	80.68%	35.00	3.00
	78.02%	80.15%	40.00	5.00
	80.69%	82.27%	37.00	3.00
	79.83%	83.13%	32.00	5.00
Mean	**79.47%**	**81.42%**	**36.40**	**3.40**
StDev	**0.98%**	**1.24%**	**3.05**	**1.67**
Alpha	**IQA Confidence Interval Width**	**Doc Confidence Interval Width**	**Mean Diff**	**Overlap (-)**
0.05	0.86%	1.08%	-1.96%	-3.90%
0.10	0.72%	0.91%	-1.96%	-3.59%
0.20	0.56%	0.71%	-1.96%	-3.23%
0.25	0.50%	0.64%	-1.96%	-3.10%
0.50	0.30%	0.37%	-1.96%	-2.62%

The positive classification roulette wheels are set with three positive terms for the following tests. Figure 2-26 shows the settings.

Search Terms	Positive Terms	Negative Terms
4,CLOSE,FRONT,GROUND,MIG-21, 0.50,TRUE 4,MEDIUM,DISTANT,INFLIGHT,MIG-21, 0.80,TRUE 4,CLOSE,FRONT,PARTIAL,MIG-21, 1.00,TRUE	3,CLOSE,FRONT,MIG-21, 0.50,TRUE 3,MEDIUM,FRONT,MIG-21, 0.80,TRUE 3,CLOSE,PARTIAL,MIG-21, 1.00,TRUE	1,MIRAGE_2000,0.50,FALSE 1,MIG-21BIS,1.00,FALSE

Figure 4-26: Probability Set Search and Classification Wheel, 4-Term-3 NAIC Search

Figure 4-27 shows a 4-Term-3 test. IQA begins with a much better accuracy due to the metadata randomization and the strength of early rule learning. Both methods quickly improve over the first 15 iterations. Generalization switches push the raw document count accuracies down, and they stabilize around 84-85%. IQA accuracy outperforms the raw document count throughout the entire test, and ends up better by 4.46% after 100 iterations.

Table 4-8 shows the results of all five tests. Metadata randomization affected early mean accuracies for both sets of 4-Term tests. These tests represent the most consistently accurate and stabile tests for any attempted. IQA outperforms the raw document count in all five tests. A 95% confidence interval test shows IQA performs better by almost 2%, and is consistently better than the raw document count at each confidence interval measurement point.

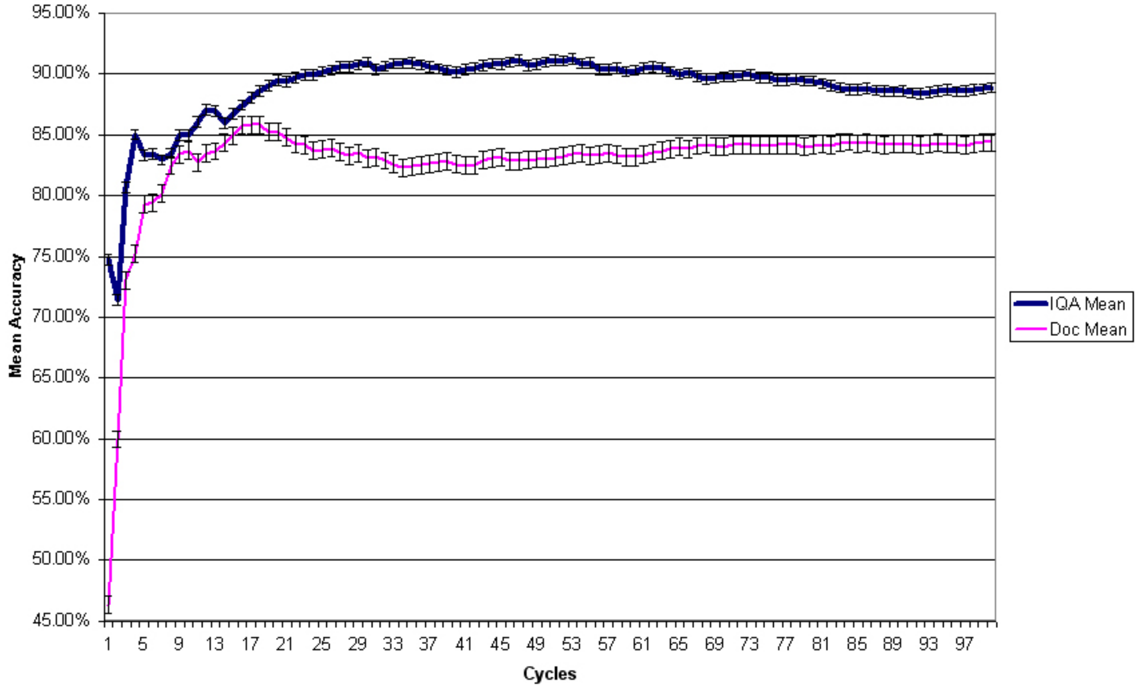


Figure 4-27: 'NAIC Dataset', 4-Term-3

Table 4-8: NAIC Test 4-Term-3

NAIC Test	IQA Mean	Doc Mean	Rules Learned	Term Generalized
4 Term 3	88.19%	86.61%	30.00	6.00
	89.43%	85.72%	27.00	4.00
	88.29%	85.79%	27.00	7.00
	88.89%	84.43%	23.00	6.00
	88.10%	84.39%	26.00	4.00
Mean	**88.58%**	**85.39%**	**26.60**	**5.40**
StDev	**0.57%**	**0.96%**	**2.51**	**1.34**
Confidence Level	**IQA Confidence Interval Width**	**Doc Confidence Interval Width**	**Mean Diff**	**Overlap**
99.00%	0.65%	1.11%	3.19%	**1.43%**
95.00%	0.50%	0.84%	3.19%	**1.85%**
90.00%	0.42%	0.71%	3.19%	**2.06%**
75.00%	0.29%	0.49%	3.19%	**2.40%**
50.00%	0.17%	0.29%	3.19%	**2.73%**

4.10 Summary

This chapter discusses each of the objectives of the IQA system, and goes though a detailed discussion on each of the functions. This analysis establishes IQA's ability to learn rules, and then generalize those rules across a defined semantic tree. We tested under automated conditions to test IQA's abilities versus a simple document count sort. The semantic structure of the data as well as the initial order of documents influences the initial performance of any search method. Appendix D shows graphs of all tests. The final chapter presents the research conclusions as well as recommendations for future work.

5. Conclusions and Recommendations

5.1 Introduction

This chapter summarizes the Intelligent Query Answering (IQA) research and research objectives. The first section discusses the impact of combining rule learning and rule generalization. The next section reviews the objectives of this research and draws conclusions regarding the efficacy of IQA. The final portion presents an outline of some potential follow on areas of study.

5.2 Research Impact

The need for better methods of effectively searching large dataset drives this research. The first objective is the construction of a rule learning system that returns documents sorted by rule weights. The second objective is to build a system that generalizes those rules with future searches. The results of this study provide an innovative method for returning relevant user requested documents by learning rules based on the way a user classifies returned documents.

5.2.1 Rule Learning

Rule learning consists of a modified implementation of FOIL combined with a Winnow algorithm used for updating rule weights. IQA learns rules based up search terms and user classification of the returned documents. It returns the documents sorted relevance weights, created by rule hits, rule weights and search term hits. The rules weights adjust over time based on user classifications of returned documents that have associated rules and rule use.

Analysis of the modified FOIL algorithm shows IQA can learn rules under all tested search conditions. The Winnow algorithm initializes rule weights for all new rules, and correctly updates rule weights on subsequent classifications of returned documents. Document searches compute the relevance weight accurately and successfully sort all returned documents based on this weight.

The complexity of the learned rules varied from one to three terms throughout the tests. However, IQA only learned three-term rules through manually testing with the 'Shape Dataset'. All automated tests generated rules of two or one terms. FOIL's greedy approach in learning rules keeps the number of rule terms to those that return positively classified documents. It is in this way that FOIL can learn two or more rules for a single search.

5.2.2 Rule Generalization

Rule generalization consists of pseudo generalization, rule promotion, rule absorption and rule aging. IQA uses existing rules and creates new pseudo rules by generalizing semantically similar terms. Pseudo rules provide an additional set of documents when rules for the existing search terms do not exist. This function assists the user finding the most relevant documents based upon previous searches (learned rules.)

Tests show that IQA's pseudo rule generalization assists in finding documents. The semantic distance threshold (δ) successfully limits rule generalization. When rules exist for the search terms, any documents returned by generalized rules have the expected lower relevance value that documents returned by existing rules. However, a generalized rule can still have a remarkably large rule weight if the semantically similar rule had a large rule weight. Testing both scenarios ensures rule weights and pseudo-generalized rule weights correctly compute the weight based on the original rule weight, δ and number of existing rules.

Tests of the rule promotion show that IQA creates a new rule from instances of existing rules when it finds two or more matching rules that are separate only by a term within the δ. The new rule is a generalized version of the previous rules (promoted.) The promotion algorithm also correctly computes the generalized rule weight based on the existing rules that matched, and then deletes those matching rules.

Rule absorption and rule aging delete rules that add no value to IQA. Rule absorption occurs when a more general form of the rule exists. Rule aging occurs when the candidate rule weight is less than 0.0125. Each function works as tested.

5.3 Automated testing

IQA runs through a series of automated tests designed to simulate a live user searching the system and compare the accuracy of returned document order. Tests use both test data sets and are completed using a negative classification wheel spin (to simulate user error) and again with no negative classification (to simulate perfect user classification.)

Figure 5-1 shows the complete ‘Shape Dataset’ results. IQA outperforms the raw document count in every simulation. IQA’s document return order is more accurate by more than 9% in some instances to that of the raw document count.

Table 5-1: ‘Shape Dataset’ Complete Results

Shape Test	IQA Mean	Doc Mean	Rules Learned	Term Generalized
2 Term 1	71.15%	65.11%	6.00	6.00
	81.23%	68.74%	6.00	6.00
	75.71%	75.79%	4.00	4.00
	82.01%	74.24%	4.00	4.00
	68.41%	68.21%	8.00	7.00
Mean	**75.70%**	**70.42%**	**5.60**	**5.40**
StDev	**6.01%**	**4.45%**	**1.67**	**1.34**
Shape Test	**IQA Mean**	**Doc Mean**	**Rules Learned**	**Term Generalized**
3 Term 1	83.87%	75.31%	6.00	6.00
	82.98%	73.30%	6.00	6.00
	89.36%	81.19%	4.00	4.00
	83.20%	76.31%	4.00	4.00
	90.50%	78.36%	8.00	7.00
Mean	**85.98%**	**76.89%**	**5.60**	**5.40**
StDev	**3.64%**	**3.01%**	**1.67**	**1.34**
Shape Test	**IQA Mean**	**Doc Mean**	**Rules Learned**	**Term Generalized**
3 Term 2	93.91%	94.16%	2.00	2.00
	81.04%	73.15%	4.00	4.00
	93.07%	92.53%	3.00	4.00
	72.58%	71.50%	3.00	3.00
	75.97%	70.02%	3.00	3.00
Mean	**83.31%**	**80.27%**	**3.00**	**3.20**
StDev	**9.77%**	**12.00%**	**0.71**	**0.84**

Table 5-2 shows the confidence intervals for the ‘Shape Dataset’ tests. While the individual tests show IQA is more accurate, the confidence tests reveal this in only consistently true in the 3-Term-1 test. In the 2-Term-1 test, there is only a 74% confidence that IQA will be more accurate that the raw document count. That confidence drops to 24% with the 3-Term-2 test. Therefore, these two tests are statistically equivalent.

Table 5-2: 'Shape Dataset' Confidence Intervals

Shape Test 2-Term-1					
Alpha	**IQA Mean**	**Doc Mean**	**Mean Diff**	**Overlap**	**Confidence**
0.05	5.27%	3.90%	5.29%	-3.88%	
0.10	4.42%	3.28%	5.29%	-2.41%	
0.20	3.44%	2.55%	5.29%	-0.71%	
0.25	3.09%	2.29%	5.29%	-0.10%	
0.26	3.03%	2.24%	5.29%	**0.02%**	**74.00%**
Shape Test 3-Term-1					
Alpha	**IQA Mean**	**Doc Mean**	**Mean Diff**	**Overlap**	**Confidence**
0.01	4.19%	3.47%	9.09%	**1.42%**	**99.00%**
0.05	3.19%	2.64%	9.09%	**3.26%**	**95.00%**
0.10	2.68%	2.22%	9.09%	**4.19%**	**90.00%**
0.25	1.87%	1.55%	9.09%	**5.67%**	**75.00%**
0.50	1.10%	0.91%	9.09%	**7.08%**	**50.00%**
Shape Test 3-Term-2					
Alpha	**IQA Mean**	**Doc Mean**	**Mean Diff**	**Overlap**	**Confidence**
0.05	8.56%	10.52%	3.04%	-16.04%	
0.10	7.19%	8.83%	3.04%	-12.97%	
0.20	5.60%	6.88%	3.04%	-9.43%	
0.25	5.03%	6.17%	3.04%	-8.15%	
0.76	1.33%	1.64%	3.04%	**0.07%**	**24.00%**

Table 5-3 shows the complete 'NAIC Dataset" test results. The raw document count outperformed with a loosely bound search (1 positive term.). However, IQA outperformed the raw document count on a tightly positively bound search. This could be due to the structure of the 'NAIC Dataset.' The NAIC data is very structured in the sense that the first few terms are always positional [CLOSE RIGHT FRONT GROUND…] followed by an object [MIG-21], followed by information on the country markings [WITH POLISH MARKINGS]. IQA makes no assumptions about this data and therefore uses none of this background knowledge to learn and generalize rules better.

The generalized terms are lower in the 'NAIC' Dataset tests than in the 'SHAPE Dataset' tests. This result is expected during testing, since a term can only be generalized if a majority of child nodes under the same parent has similar rules. The automated testing exploited only a few of these possible generalizations, and this in turn could have an affect on the overall IQA accuracy results.

Table 5-3: 'NAIC Dataset' Complete Results

NAIC Test	IQA Mean	Doc Mean	Rules Learned	Term Generalized
4 Term 1	79.16%	80.88%	38.00	1.00
	79.62%	80.68%	35.00	3.00
	78.02%	80.15%	40.00	5.00
	80.69%	82.27%	37.00	3.00
	79.83%	83.13%	32.00	5.00
Mean	**79.47%**	**81.42%**	**36.40**	**3.40**
StDev	**0.98%**	**1.24%**	**3.05**	**1.67**
NAIC Test	**IQA Mean**	**Doc Mean**	**Rules Learned**	**Term Generalized**
4 Term 3	88.19%	86.61%	30.00	6.00
	89.43%	85.72%	27.00	4.00
	88.29%	85.79%	27.00	7.00
	88.89%	84.43%	23.00	6.00
	88.10%	84.39%	26.00	4.00
Mean	**88.58%**	**85.39%**	**26.60**	**5.40**
StDev	**0.57%**	**0.96%**	**2.51**	**1.34**

Table 5-3: 'NAIC Dataset' Complete Results

Table 5-4 shows the confidence intervals for the 'NAIC Dataset' tests. The results are inconsistent between the two tests The 4-Term-1 test shows that the IQA and raw document counts are statistically equivalent, while the 4-Term-3 test shows IQA performs a bit better than the raw document count.

Overall, IQA outperformed the raw document count in four out of five tests. Confidence intervals conclude that in three of these tests results are statistically the same, while in the other two IQA performs somewhat better than the raw document count. Manual testing shows the effectiveness of rule learning and generalization, while the automated testing confirms that it is no worse than a raw document count in all instances. In some scenarios, IQA outperforms the raw document count.

Table 5-4: 'NAIC Dataset' Confidence Intervals

NAIC Test 4-Term-1					
Alpha	**IQA Mean**	**Doc Mean**	**Mean Diff**	**Overlap**	**Confidence**
0.05	0.86%	1.08%	-1.96%	-3.90%	
0.10	0.72%	0.91%	-1.96%	-3.59%	
0.20	0.56%	0.71%	-1.96%	-3.23%	
0.25	0.50%	0.64%	-1.96%	-3.10%	
0.50	0.30%	0.37%	-1.96%	-2.62%	
NAIC Test 4-Term-3					
Alpha	**IQA Mean**	**Doc Mean**	**Mean Diff**	**Overlap**	**Confidence**
0.01	0.65%	1.11%	3.19%	**1.43%**	**99.00%**
0.10	0.42%	0.71%	3.19%	**2.06%**	**90.00%**
0.20	0.33%	0.55%	3.19%	**2.31%**	**80.00%**
0.25	0.29%	0.49%	3.19%	**2.40%**	**75.00%**
0.50	0.17%	0.29%	3.19%	**2.73%**	**50.00%**

5.4 Outline of Future Work

This thesis provides a fundamental look at the concepts of blending rule learning and rule generalization, as well as providing the foundation for future research in this area. The concepts behind IQA have relevance in almost any database or web search application. Those concepts could improve an individual or group of individuals' abilities to get relevant result from a large dataset after on a few search iterations. This section also discusses a more robust implementation using group rule sets, a semantic tree builder/learner, database implementation, and a WordNet plug in.

5.4.1 Local versus Group Rule Pseudo Generalization

IQA learns rules for one user. It can be expanded in such a way to provide a new user access to a global set of strong rules to pseudo generalize against for a limited time, or until a threshold if personal rules were established. This global rule set would provide an initial basis for a user to search a large data structure, and has the potential for providing relevant documents more effectively than individual rule learning.

5.4.2 Semantic Tree Builder

The semantic tree forms the basis for rule generalization. The process of manually building an effective semantic tree is tedious. An alternative is the development of a semantic tree builder that builds a list of all terms from the database, and then allows the user to build the tree interactively. Another method would be to learn the semantic tree structure dynamically from user input.

5.4.3 Database Implementation

For the scope of this research, the Java data structures were adequate to support the implementation and test of IQA at a rudimentary level. To implement these concepts on a larger scale, the rule learning and generalization aspects need incorporation into an ODCB compliant structure such as SQL-Server or Oracle to support large data sets. The NAIC IEC system has been the subject of many research efforts [BAK03, WIL03 and SPL04] and it is possible that one of these approaches could benefit from integration with IQA.

5.4.4 WordNet Interface

One of this research challenges is building the semantic tree. WordNet provided the concepts for IQA's semantic tree and aided with term placement. However, to incorporate WordNet into IQA would have required a complete rewrite of both applications. Still, future research to incorporate WordNet into an open structure would be valuable. Such a plug and play structure would allow future machine learning approaches to searching databases using rule generalization without continuously reinventing the wheel.

5.5 Summary

This research examines the problem of querying a database with large amounts of information and returning only the most relevant records to the user. Specifically, the problem is the effective retrieval of desired records in a relevant order without a tremendous amount of data preprocessing. The research integrates one popular approach, rule learning using FOIL, with the more obscure concept of semantic generalization. The result is a rule learning system that can also generalize rules across a predefined semantic tree. This research provides a first look at this novel combination and demonstrates the capabilities against two sets of data. It also provides ideas for future studies to extend these concepts.

Appendix A – Semantic Trees

1. Raw 'Shape Dataset' Semantic Data

```
WORD,null,NOUN,ADJECTIVE
NOUN,WORD,SHAPE
ADJECTIVE,WORD,ATTRIBUTE
SHAPE,NOUN,CIRCLE,SQUARE,TRIANGLE
CIRCLE,SHAPE,null
SQUARE,SHAPE,null
TRIANGLE,SHAPE,null
ATTRIBUTE,ADJECTIVE,SIZE,COLOR
SIZE,ATTRIBUTE,SMALL,MEDIUM,BIG
SMALL,SIZE,null
MEDIUM,SIZE,null
BIG,SIZE,null
COLOR,ATTRIBUTE,RED,BLUE,GREEN
RED,COLOR,null
BLUE,COLOR,null
GREEN,COLOR,null
```

2. Raw 'NAIC Dataset' Semantic Data

```
WORD,null,OBJECT,DESCRIPTOR
OBJECT,WORD,AIRCRAFT,SPACE_VEHICLE,GUIDED_MISSILES
AIRCRAFT,OBJECT,MANNED,UAV
MANNED,AIRCRAFT,A-47,A5C,A-5III,AN-26,ALPHA_JET,AN-12,AN-124,AN-140,AN-
225,AN-225,AN-26,AN-3,AN-32,AN-71,AN-74,AN-74-300,AN-74T-200,AN-74TK-
200,ANTONOV,B707,C-160,CANBERRA,CHEETAH-C,CHEETAH-D,DORNIER-
228,EUROFIGHTER,TYPHOON,F-15,F-16,F-2,F-4,F-6,F-7MF,F-7MG,F-8IIACT,F-
8IIM,FB-7,FBC-1,FC-1,FT-7PG,FTC-2000,HAL,JAGUAR,JAS-39,K-8,KA-28,KA-
52,KMH,KT-1,L15,L-159,L-39,LCA,MB-326,MI-17-V5,MIG-21,MIG-27M,MIG-
29,MIRAGE_2000,MIRAGE_F1CR,MIRAGE_F1CT,PC-12,RAFALE,SU-22,SU-22M4,SU-
24MK,SU-27FLANKER,SU-30MK,SU-32,SU-33,SU-39,ETENDARD,MK-III,T-
50,TORNADO,TU-334,XXJ,Y7H-500,Y8F400,YAK-130,Z-8,Z-9G
A-47,MANNED,DAKOTA
DAKOTA,A-47,null
A5C,MANNED,null
A-5III,MANNED,FANTANS
FANTANS,A-5III,null
AN-26,MANNED,null
ALPHA_JET,MANNED,null
AN-12,MANNED,CUB
CUB,AN-12,null
AN-124,MANNED,CONDOR
CONDOR,AN-124,null
AN-140,MANNED,null
AN-225,MANNED,null
AN-225,MANNED,COSSACK
COSSACK,AN-225,null
AN-26,MANNED,CURL
```

```
CURL,AN-26,null
AN-3,MANNED,null
AN-32,MANNED,CLINE,AN-32B,AN-32P
CLINE,AN-32,null
AN-32B,AN-32,null
AN-32P,AN-32,null
AN-71,MANNED,null
AN-74,MANNED,null
AN-74-300,MANNED,null
AN-74T-200,MANNED,null
AN-74TK-200,MANNED,null
ANTONOV,MANNED,null
B707,MANNED,COMINT,COMJAM
COMINT,B707,null
COMJAM,B707,null
C-160,MANNED,GABRIEL
GABRIEL,C-160,null
CANBERRA,MANNED,null
CHEETAH-C,MANNED,null
CHEETAH-D,MANNED,null
DORNIER-228,MANNED,null
EUROFIGHTER,MANNED,null
TYPHOON,MANNED,null
F-15,MANNED,null
F-16,MANNED,F-16A
F-16A,F-16,null
F-2,MANNED,null
F-4,MANNED,PHANTOM
PHANTOM,F-4,null
F-6,MANNED,null
F-7MF,MANNED,null
F-7MG,MANNED,null
F-8IIACT,MANNED,null
F-8IIM,MANNED,null
FB-7,MANNED,null
FBC-1,MANNED,null
FC-1,MANNED,null
FT-7PG,MANNED,null
FTC-2000,MANNED,null
HAL,MANNED,null
JAGUAR,MANNED,null
JAS-39,MANNED,GRIPEN,JAS-39A
GRIPEN,JAS-39,null
JAS-39A,JAS-39,null
K-8,MANNED,null
KA-28,MANNED,KAMOV
KAMOV,KA-28,null
KA-52,MANNED,ALLIGATOR
ALLIGATOR,KA-52,null
KMH,MANNED,null
KT-1,MANNED,WOONGBEE
WOONGBEE,KT-1,null
L15,MANNED,null
L-159,MANNED,null
L-39,MANNED,null
```

```
LCA,MANNED,LCA-NAVY
LCA-NAVY,LCA,null
MB-326,MANNED,IMPALA
IMPALA,MB-326,null
MI-17-V5,MANNED,null
MIG-21,MANNED,MIG-21BIS,FISHBED,MIG-21-93,MIG-21MF,MIG-21UM
BIS,MIG-21,null
FISHBED,MIG-21,null
MIG-21-93,MIG-21,null
MIG-21BIS,MIG-21,null
MIG-21MF,MIG-21,null
MIG-21UM,MIG-21,null
MIG-27M,MANNED,null
MIG-29,MANNED,null
MIRAGE_2000,MANNED,MIRAGE_2000-5F,MIRAGE_2000D
MIRAGE_2000-5F,MIRAGE_2000,null
MIRAGE_2000D,MIRAGE_2000,null
MIRAGE_F1CR,MANNED,null
MIRAGE_F1CT,MANNED,null
PC-12,MANNED,null
RAFALE,MANNED,RAFALE_B,RAFALE_B-01,RAFALE_M,RAFALE_M-02,RAFALE_M9
RAFALE_B,RAFALE,null
RAFALE_B-01,RAFALE,null
RAFALE_M,RAFALE,null
RAFALE_M-02,RAFALE,null
RAFALE_M9,RAFALE,null
SU-22,MANNED,SU-22M3
SU-22M3,SU-22,null
SU-22M4,MANNED,FITTER-K
FITTER-K,SU-22M4,null
SU-24MK,MANNED,null
SU-27FLANKER,MANNED,FLANKER,SU-27SK,FLANKER-B
FLANKER,SU-27,null
SU-27SK,SU-27,null
FLANKER-B,SU-27,null
SU-30MK,MANNED,null
SU-32,MANNED,null
SU-33,MANNED,null
SU-39,MANNED,null
ETENDARD,MANNED,null
MK-III,MANNED,SUPERHIND
SUPERHIND,MK-III,null
T-50,MANNED,null
TORNADO,MANNED,GR-4,ILS
GR-4,TORNADO,null
ILS,TORNADO,null
TU-334,MANNED,null
XXJ,MANNED,null
Y7H-500,MANNED,null
Y8F400,MANNED,null
YAK-130,MANNED,null
Z-8,MANNED,null
Z-9G,MANNED,null
UAV,AIRCRAFT,350ENGINE,AERONEF,AEROSKY,AEROSONDE,AW-
4,BREZEL/KZO,C22,CHUNGSHYANG-II,CK1G,EAGLE,FOXAT3,FOXMLCS,HERMES,MINI-
```

```
V,HERMES1500,HERMES180,HW-
02,LARK,MICRO,NISHANT,PETITDUC,PHANTOMEYE,REMEZ-3,S-100,S-45,S-70,SA-
6,SAABSHARC,SCOUT-II,SEAMOS,SEEKER,SEEKER-II,SHADOW-
200,SHARC,SKUA,TAILSITTER,VULTURE,W-50,WZ-2000
350ENGINE,UAV,null
AERONEF,UAV,null
AEROSKY,UAV,null
AEROSONDE,UAV,null
AW-4,UAV,SHARK-II
SHARK-II,AW-4,null
BREZEL/KZO,UAV,null
C22,UAV,null
CHUNGSHYANG-II,UAV,null
CK1G,UAV,null
EAGLE,UAV,null
FOXAT3,UAV,null
FOXMLCS,UAV,null
HERMES,UAV,null
MINI-V,UAV,null
HERMES1500,UAV,null
HERMES180,UAV,null
HW-02,UAV,null
LARK,UAV,null
MICRO,UAV,null
NISHANT,UAV,null
PETITDUC,UAV,null
PHANTOMEYE,UAV,null
REMEZ-3,UAV,null
S-100,UAV,null
S-45,UAV,null
S-70,UAV,null
SA-6,UAV,null
SAABSHARC,UAV,null
SCOUT-II,UAV,null
SEAMOS,UAV,null
SEEKER,UAV,null
SEEKER-II,UAV,null
SHADOW-200,UAV,null
SHARC,UAV,null
SKUA,UAV,null
TAILSITTER,UAV,null
VULTURE,UAV,null
W-50,UAV,null
WZ-2000,UAV,null
GUIDED_MISSILES,OBJECT,LGB,AA-10,AA-11,AA-12,AA-8,A-DARTER,R-
DARTER,UMKHONTO-IR,AGM-65,APS-784,AS-10,AS-11,AS-12,AS-15B,AS-
17,KRYPTON,AS-18,AS-20,KAYAK,ASTER-30,BRAHMOS,C-301,C-701,C-
802,CINGOZ,CK1G,CROTALE,DZ-88,FLG-1,FLS-1,FLV-1,FM-90N,FN-6,HJ-8,HN-
5,HQ-2B,I1-2000,INGWE,I-TALD,JL1,KEPD-350,KH-59MK,KS-1A,LY-60,MAGIC-
2,METIS-M,MICA,MK-80,MM-2000,MOKOPA,MOSKIT-E,PAVEWAY-III,MK2,PL-5E,PL-
9C,TY-90,QW-1,QW-3,QW-2,QW-3,QW-3,QW-4,RAPTOR-I,RAPTOR-II,SA-10,SA-
16,SA-6,SAHV-3,SAHV-IR,SAMOC,STORM,SHADOW/SCALPEG,TIENCHIEN-II,TY-
90,PL-5E,PL-9C,UA-424,UMKHONTO,UMKHONTO-IR,A-DARTER,WS-1
LGB,GUIDED_MISSILES,null
AA-10,GUIDED_MISSILES,ALAMO
```

```
ALAMO,AA-10,null
AA-11,GUIDED_MISSILES,null
AA-12,GUIDED_MISSILES,null
AA-8,GUIDED_MISSILES,null
A-DARTER,GUIDED_MISSILES,null
R-DARTER,GUIDED_MISSILES,null
UMKHONTO-IR,GUIDED_MISSILES,null
AGM-65,GUIDED_MISSILES,AGM-65D,AGM-65E,AGM-65B,AGM-65F
AGM-65D,AGM-65,null
AGM-65E,AGM-65,null
AGM-65B,AGM-65,null
AGM-65F,AGM-65,null
APS-784,GUIDED_MISSILES,null
AS-10,GUIDED_MISSILES,KAREN
KAREN,AS-10,null
AS-11,GUIDED_MISSILES,KILTER
KILTER,AS-11,null
AS-12,GUIDED_MISSILES,KEGLER
KEGLER,AS-12,null
AS-15B,GUIDED_MISSILES,KENT
KENT,AS-15B,null
AS-17,GUIDED_MISSILES,KRYPTON
KRYPTON,GUIDED_MISSILES,null
AS-18,GUIDED_MISSILES,KAZOO,AS-18M
KAZOO,AS-18,null
AS-18M,AS-18,null
AS-20,GUIDED_MISSILES,KAYAK
KAYAK,AS-20,null
ASTER-30,GUIDED_MISSILES,null
BRAHMOS,GUIDED_MISSILES,null
C-301,GUIDED_MISSILES,null
C-701,GUIDED_MISSILES,null
C-802,GUIDED_MISSILES,null
CINGOZ,GUIDED_MISSILES,null
CK1G,GUIDED_MISSILES,null
CROTALE,GUIDED_MISSILES,null
DZ-88,GUIDED_MISSILES,null
FLG-1,GUIDED_MISSILES,null
FLS-1,GUIDED_MISSILES,null
FLV-1,GUIDED_MISSILES,null
FM-90N,GUIDED_MISSILES,null
FN-6,GUIDED_MISSILES,null
HJ-8,GUIDED_MISSILES,RED_ARROW
RED_ARROW,HJ-8,null
HN-5,GUIDED_MISSILES,null
HQ-2B,GUIDED_MISSILES,null
I1-2000,GUIDED_MISSILES,null
INGWE,GUIDED_MISSILES,null
I-TALD,GUIDED_MISSILES,null
JL1,GUIDED_MISSILES,null
KEPD-350,GUIDED_MISSILES,null
KH-59MK,GUIDED_MISSILES,null
KS-1A,GUIDED_MISSILES,null
LY-60,GUIDED_MISSILES,null
MAGIC-2,GUIDED_MISSILES,null
```

```
METIS-M,GUIDED_MISSILES,null
MICA,GUIDED_MISSILES,null

MK-80,GUIDED_MISSILES,null
MM-2000,GUIDED_MISSILES,null
MOKOPA,GUIDED_MISSILES,null
MOSKIT-E,GUIDED_MISSILES,null
PAVEWAY-III,GUIDED_MISSILES,null
MK2,GUIDED_MISSILES,PENGUIN
PENGUIN,MK2,null
PL-5E,GUIDED_MISSILES,null
PL-9C,GUIDED_MISSILES,null
TY-90,GUIDED_MISSILES,null
QW-1,GUIDED_MISSILES,QW-1A,QW-1M
QW-1A,QW-1,null
QW-1M,QW-1,null
QW-3,GUIDED_MISSILES,null
QW-2,GUIDED_MISSILES,null
QW-3,GUIDED_MISSILES,null
QW-3,GUIDED_MISSILES,null
QW-4,GUIDED_MISSILES,null
RAPTOR-I,GUIDED_MISSILES,null
RAPTOR-II,GUIDED_MISSILES,null
SA-10,GUIDED_MISSILES,null
SA-16,GUIDED_MISSILES,GIMLET
GIMLET,SA-16,null
SA-6,GUIDED_MISSILES,null
SAHV-3,GUIDED_MISSILES,null
SAHV-IR,GUIDED_MISSILES,null
SAMOC,GUIDED_MISSILES,null
STORM,GUIDED_MISSILES,null

SHADOW/SCALPEG,GUIDED_MISSILES,null
TIENCHIEN-II,GUIDED_MISSILES,null
TY-90,GUIDED_MISSILES,null
PL-5E,GUIDED_MISSILES,null
PL-9C,GUIDED_MISSILES,null
UA-424,GUIDED_MISSILES,null
UMKHONTO,GUIDED_MISSILES,null
UMKHONTO-IR,GUIDED_MISSILES,null
A-DARTER,GUIDED_MISSILES,null
WS-1,GUIDED_MISSILES,WS-1B
WS-1B,WS-1,null
SPACE_VEHICLE,OBJECT,CYCLONE-4,ZENIT-3SL,CZ-2E,CZ-2F,CZ-3A,CZ-3B,CZ-
3C,DFH-1,HANGTIAN_TSINGHUA-1,HY-1,LM-2C/SD,LM-2E,LM-3A,LM-3B,KT-1,KT-
2,SHENZHOU-1
CYCLONE-4,SPACE_VEHICLE,null
ZENIT-3SL,SPACE_VEHICLE,null
CZ-2E,SPACE_VEHICLE,LONGMARCH
LONGMARCH,CZ-2E,null
CZ-2F,SPACE_VEHICLE,null
CZ-3A,SPACE_VEHICLE,null
CZ-3B,SPACE_VEHICLE,null
CZ-3C,SPACE_VEHICLE,null
DFH-1,SPACE_VEHICLE,null
```

```
HANGTIAN_TSINGHUA-1,SPACE_VEHICLE,null
HY-1,SPACE_VEHICLE,OCEAN
OCEAN,HY-1,nullA
LM-2C/SD,SPACE_VEHICLE,null
LM-2E,SPACE_VEHICLE,null
LM-3A,SPACE_VEHICLE,null
LM-3B,SPACE_VEHICLE,null
KT-1,SPACE_VEHICLE,PIONEER-1
PIONEER-1,KT-1,null
KT-2,SPACE_VEHICLE,PIONEER-2,PIONEER-2A
PIONEER-2,KT-2,null

PIONEER-2A,KT-2,null
SHENZHOU-1,SPACE_VEHICLE,null
DESCRIPTOR,WORD,DISTANCE,POSITION,LOCATION,VIEW,MARKINGS
CLOSE,DISTANCE,null
MEDIUM,DISTANCE,null
DISTANT,DISTANCE,,null
DISTANCE,DESCRIPTOR,CLOSE,MEDIUM,DISTANT
POSITION,DESCRIPTOR,LEFT,RIGHT
LEFT,POSITION,null
RIGHT,POSITION,null
LOCATION,DESCRIPTOR,FRONT,REAR,INTERIOR,SIDE,UNDERSIDE,OVERHEAD,AMOUNT
FRONT,LOCATION,null
REAR,LOCATION,null
INTERIOR,LOCATION,null
SIDE,LOCATION,null
UNDERSIDE,LOCATION,null
OVERHEAD,LOCATION,null
VIEW,DESCRIPTOR,GROUND,INFLIGHT
GROUND,VIEW,null
INFLIGHT,VIEW,null
AMOUNT,DESCRIPTOR,PARTIAL,FULL
PARTIAL,AMOUNT,null
FULL,AMOUNT,null
MARKINGS,DESCRIPTOR,PAKISTANI,UKRAINIAN,PERUVIAN,AFRICAN,UK,US,KOREAN,J
APANESE,TURKISH,FRENCH,SWEDISH,RUSSIAN,YEMEN,INDIAN,CZECH,POLISH,YUGOSL
AVIAN,BANGLADESH,UGANDAN,CHINESE,GERMAN,RSAF
PAKISTANI,MARKINGS,null
UKRAINIAN,MARKINGS,null
PERUVIAN,MARKINGS,null
AFRICAN,MARKINGS,null
UK,MARKINGS,null
US,MARKINGS,null
KOREAN,MARKINGS,null
JAPANESE,MARKINGS,null
TURKISH,MARKINGS,null
FRENCH,MARKINGS,null
SWEDISH,MARKINGS,null
RUSSIAN,MARKINGS,null
YEMEN,MARKINGS,null
INDIAN,MARKINGS,null
CZECH,MARKINGS,null
POLISH,MARKINGS,null
YUGOSLAVIAN,MARKINGS,null
```

```
BANGLADESH,MARKINGS,null
UGANDAN,MARKINGS,null
CHINESE,MARKINGS,null
GERMAN,MARKINGS,null
RSAF,MARKINGS,null
```

Appendix B – Winnow Rule Aging Calculations

	Initial Classification	Initial Classification	1 Pos	2 Pos	3 Pos	4 Pos	5 Pos	6 Pos
Subsequent Iterations	**1.00000**	**1.00000**	**1.50000**	**2.25000**	**3.37500**	**5.06250**	**7.59375**	**11.39063**
1	0.90000	***0.50000***[2]	1.35000	2.02500	***1.68750***	***2.53125***	***3.79688***	***5.69531***
2	0.81000	***0.25000***	1.21500	1.82250	***0.84375***	***1.26563***	***1.89844***	***2.84766***
3	0.72900	***0.12500***	1.09350	1.64025	0.75938	***0.63281***	***0.94922***	***1.42383***
4	0.65610	***0.06250***	0.98415	1.47623	0.68344	***0.31641***	***0.47461***	***0.71191***
5	0.59049	***0.03125***	0.88574	1.32860	0.61509	0.28477	***0.23730***	***0.35596***
6	0.53144	***0.01563***	0.79716	1.19574	0.55358	0.25629	***0.11865***	***0.17798***
7	0.47830	**0.00781**	0.71745	1.07617	0.49823	0.23066	0.10679	***0.08899***
8	0.43047	0.00703	0.64570	0.96855	0.44840	0.20759	0.09611	***0.04449***
9	0.38742	0.00633	0.58113	0.87170	0.40356	0.18683	0.08650	0.04005
10-17								
18	0.15009	0.00245	0.22514	0.33771	0.15635	0.07238	0.03351	0.01551
19	0.13509	0.00221	0.20263	0.30394	0.14071	0.06515	0.03016	**0.01396**
20	0.12158	0.00199	0.18236	0.27355	0.12664	0.05863	0.02714	0.01257
21	0.10942	0.00179	0.16413	0.24619	0.11398	0.05277	0.02443	0.01131
22	0.09848	0.00161	0.14772	0.22157	0.10258	0.04749	0.02199	0.01018
23	0.08863	0.00145	0.13294	0.19942	0.09232	0.04274	0.01979	0.00916
24	0.07977	0.00130	0.11965	0.17947	0.08309	0.03847	0.01781	0.00824
25	0.07179	0.00117	0.10768	0.16153	0.07478	0.03462	0.01603	0.00742
26	0.06461	0.00106	0.09692	0.14537	0.06730	0.03116	**0.01443**	0.00668
27	0.05815	0.00095	0.08722	0.13084	0.06057	0.02804	0.01298	0.00601
28	0.05233	0.00085	0.07850	0.11775	0.05452	0.02524	0.01168	0.00541
29	0.04710	0.00077	0.07065	0.10598	0.04906	0.02271	0.01052	0.00487
30	0.04239	0.00069	0.06359	0.09538	0.04416	0.02044	0.00946	0.00438
31	0.03815	0.00062	0.05723	0.08584	0.03974	0.01840	0.00852	0.00394
32	0.03434	0.00056	0.05151	0.07726	0.03577	**0.01656**	0.00767	0.00355
33	0.03090	0.00050	0.04635	0.06953	0.03219	0.01490	0.00690	0.00319
34	0.02781	0.00045	0.04172	0.06258	0.02897	0.01341	0.00621	0.00287
35	0.02503	0.00041	0.03755	0.05632	0.02607	0.01207	0.00559	0.00259
36	0.02253	0.00037	0.03379	0.05069	0.02347	0.01086	0.00503	0.00233
37	0.02028	0.00033	0.03041	0.04562	0.02112	0.00978	0.00453	0.00210
38	0.01825	0.00030	0.02737	0.04106	0.01901	0.00880	0.00407	0.00189
39	0.01642	0.00027	0.02463	0.03695	0.01711	0.00792	0.00367	0.00170
40	**0.01478**[3]	0.00024	0.02217	0.03326	0.01540	0.00713	0.00330	0.00153
41	0.01330	0.00022	0.01995	0.02993	**0.01386**	0.00642	0.00297	0.00138
42	0.01197	0.00020	0.01796	0.02694	0.01247	0.00577	0.00267	0.00124
43	0.01078	0.00018	0.01616	0.02424	0.01122	0.00520	0.00241	0.00111
44	0.00970	0.00016	**0.01455**	0.02182	0.01010	0.00468	0.00217	0.00100
45	0.00873	0.00014	0.01309	0.01964	0.00909	0.00421	0.00195	0.00090
46	0.00786	0.00013	0.01178	0.01767	0.00818	0.00379	0.00175	0.00081
47	0.00707	0.00012	0.01060	0.01591	0.00736	0.00341	0.00158	0.00073
48	0.00636	0.00010	0.00954	**0.01432**	0.00663	0.00307	0.00142	0.00066

[2] Items in bold italics indicate negative classification, and subsequent factoring of 0.5.

[3] Items in bold indicate non-classification, and subsequent factoring of 0.9.

Appendix C – Log File: IQA Tests Sections 4.4 – 4.5

```
======================================================
This file contains the comparison data for testing.
======================================================
>>>>> Search Results <<<<<
======================================================
Total number of search results = 24
======================================================
[[SMALL, TRIANGLE], [BLUE, TRIANGLE], 1.0]
[[SMALL, TRIANGLE], [SMALL, GREEN, SQUARE], 1.0]
[[SMALL, TRIANGLE], [SMALL, CIRCLE], 1.0]
[[SMALL, TRIANGLE], [BIG, GREEN, TRIANGLE], 1.0]
[[SMALL, TRIANGLE], [RED, TRIANGLE], 1.0]
[[SMALL, TRIANGLE], [BIG, BLUE, TRIANGLE], 1.0]
[[SMALL, TRIANGLE], [GREEN, TRIANGLE], 1.0]
[[SMALL, TRIANGLE], [SMALL, GREEN, CIRCLE], 1.0]
[[SMALL, TRIANGLE], [MEDIUM, TRIANGLE], 1.0]
[[SMALL, TRIANGLE], [SMALL, SQUARE], 1.0]
[[SMALL, TRIANGLE], [MEDIUM, GREEN, TRIANGLE], 1.0]
[[SMALL, TRIANGLE], [BIG, RED, TRIANGLE], 1.0]
[[SMALL, TRIANGLE], [SMALL, RED, CIRCLE], 1.0]
[[SMALL, TRIANGLE], [TRIANGLE], 1.0]
[[SMALL, TRIANGLE], [BIG, TRIANGLE], 1.0]
[[SMALL, TRIANGLE], [SMALL, BLUE, CIRCLE], 1.0]
[[SMALL, TRIANGLE], [SMALL, BLUE, TRIANGLE], 2.0]
[[SMALL, TRIANGLE], [MEDIUM, RED, TRIANGLE], 1.0]
[[SMALL, TRIANGLE], [MEDIUM, BLUE, TRIANGLE], 1.0]
[[SMALL, TRIANGLE], [SMALL, TRIANGLE], 2.0]
[[SMALL, TRIANGLE], [SMALL, RED, TRIANGLE], 2.0]
[[SMALL, TRIANGLE], [SMALL, BLUE, SQUARE], 1.0]
[[SMALL, TRIANGLE], [SMALL, GREEN, TRIANGLE], 2.0]
[[SMALL, TRIANGLE], [SMALL, RED, SQUARE], 1.0]
******************************************************
======================================================
>>>>> UnclassifiedResults <<<<<
======================================================
Total number of unclassified results = 24
======================================================
[[SMALL, TRIANGLE], [SMALL, BLUE, TRIANGLE], 2.0]
[[SMALL, TRIANGLE], [SMALL, TRIANGLE], 2.0]
[[SMALL, TRIANGLE], [SMALL, RED, TRIANGLE], 2.0]
[[SMALL, TRIANGLE], [SMALL, GREEN, TRIANGLE], 2.0]
[[SMALL, TRIANGLE], [RED, TRIANGLE], 1.0]
[[SMALL, TRIANGLE], [BIG, BLUE, TRIANGLE], 1.0]
[[SMALL, TRIANGLE], [GREEN, TRIANGLE], 1.0]
[[SMALL, TRIANGLE], [SMALL, GREEN, CIRCLE], 1.0]
[[SMALL, TRIANGLE], [MEDIUM, TRIANGLE], 1.0]
[[SMALL, TRIANGLE], [SMALL, SQUARE], 1.0]
[[SMALL, TRIANGLE], [MEDIUM, GREEN, TRIANGLE], 1.0]
[[SMALL, TRIANGLE], [BIG, RED, TRIANGLE], 1.0]
[[SMALL, TRIANGLE], [SMALL, RED, CIRCLE], 1.0]
[[SMALL, TRIANGLE], [TRIANGLE], 1.0]
```

```
[[SMALL, TRIANGLE], [BIG, TRIANGLE], 1.0]
[[SMALL, TRIANGLE], [SMALL, BLUE, CIRCLE], 1.0]
[[SMALL, TRIANGLE], [BLUE, TRIANGLE], 1.0]
[[SMALL, TRIANGLE], [MEDIUM, RED, TRIANGLE], 1.0]
[[SMALL, TRIANGLE], [MEDIUM, BLUE, TRIANGLE], 1.0]
[[SMALL, TRIANGLE], [SMALL, GREEN, SQUARE], 1.0]
[[SMALL, TRIANGLE], [SMALL, CIRCLE], 1.0]
[[SMALL, TRIANGLE], [SMALL, BLUE, SQUARE], 1.0]
[[SMALL, TRIANGLE], [BIG, GREEN, TRIANGLE], 1.0]
[[SMALL, TRIANGLE], [SMALL, RED, SQUARE], 1.0]
******************************************************
======================================================
>>>>> Classification Results <<<<<
======================================================
Total number of classified results = 24
======================================================
[[SMALL, TRIANGLE], [SMALL, BLUE, TRIANGLE], 0]
[[SMALL, TRIANGLE], [SMALL, TRIANGLE], 0]
[[SMALL, TRIANGLE], [SMALL, RED, TRIANGLE], 1]
[[SMALL, TRIANGLE], [SMALL, GREEN, TRIANGLE], 0]
[[SMALL, TRIANGLE], [RED, TRIANGLE], 0]
[[SMALL, TRIANGLE], [BIG, BLUE, TRIANGLE], 0]
[[SMALL, TRIANGLE], [GREEN, TRIANGLE], 0]
[[SMALL, TRIANGLE], [SMALL, GREEN, CIRCLE], 0]
[[SMALL, TRIANGLE], [MEDIUM, TRIANGLE], 0]
[[SMALL, TRIANGLE], [SMALL, SQUARE], 0]
[[SMALL, TRIANGLE], [MEDIUM, GREEN, TRIANGLE], 0]
[[SMALL, TRIANGLE], [BIG, RED, TRIANGLE], 0]
[[SMALL, TRIANGLE], [SMALL, RED, CIRCLE], 0]
[[SMALL, TRIANGLE], [TRIANGLE], 0]
[[SMALL, TRIANGLE], [BIG, TRIANGLE], 0]
[[SMALL, TRIANGLE], [SMALL, BLUE, CIRCLE], 0]
[[SMALL, TRIANGLE], [BLUE, TRIANGLE], 0]
[[SMALL, TRIANGLE], [MEDIUM, RED, TRIANGLE], 0]
[[SMALL, TRIANGLE], [MEDIUM, BLUE, TRIANGLE], 0]
[[SMALL, TRIANGLE], [SMALL, GREEN, SQUARE], 0]
[[SMALL, TRIANGLE], [SMALL, CIRCLE], 0]
[[SMALL, TRIANGLE], [SMALL, BLUE, SQUARE], 0]
[[SMALL, TRIANGLE], [BIG, GREEN, TRIANGLE], 0]
[[SMALL, TRIANGLE], [SMALL, RED, SQUARE], 0]
******************************************************
======================================================
>>>>>  NEW RULES <<<<<
======================================================
Total number of new Rules = 1
======================================================
[[SMALL, TRIANGLE], [RED, SMALL, TRIANGLE], 1.0]
******************************************************
======================================================
>>>>>  DOCUMENTS FOR COMPARISON TO FOIL/WINNOW <<<<<
======================================================
Total number of documents in document file = 0
======================================================
******************************************************
======================================================
```

```
>>>>>  RULES BEFORE PROMOTION <<<<<
======================================================
Total number of Rules = 1
======================================================
[[SMALL, TRIANGLE], [RED, SMALL, TRIANGLE], 1.0]
******************************************************
======================================================
>>>>>  RULES AFTER PROMOTION <<<<<
======================================================
Total number of Rules = 1
======================================================
[[SMALL, TRIANGLE], [RED, SMALL, TRIANGLE], 1.0]
******************************************************
======================================================
This file contains the comparison data for testing.
======================================================
>>>>> Search Results <<<<<
======================================================
Total number of search results = 24
======================================================
[[RED, TRIANGLE], [BLUE, TRIANGLE], 1.0]
[[RED, TRIANGLE], [BIG, GREEN, TRIANGLE], 1.0]
[[RED, TRIANGLE], [RED, TRIANGLE], 2.0]
[[RED, TRIANGLE], [BIG, BLUE, TRIANGLE], 1.0]
[[RED, TRIANGLE], [GREEN, TRIANGLE], 1.0]
[[RED, TRIANGLE], [MEDIUM, TRIANGLE], 1.0]
[[RED, TRIANGLE], [MEDIUM, GREEN, TRIANGLE], 1.0]
[[RED, TRIANGLE], [BIG, RED, CIRCLE], 1.0]
[[RED, TRIANGLE], [BIG, RED, TRIANGLE], 2.0]
[[RED, TRIANGLE], [MEDIUM, RED, SQUARE], 1.0]
[[RED, TRIANGLE], [SMALL, RED, CIRCLE], 1.0]
[[RED, TRIANGLE], [TRIANGLE], 1.0]
[[RED, TRIANGLE], [BIG, TRIANGLE], 1.0]
[[RED, TRIANGLE], [BIG, RED, SQUARE], 1.0]
[[RED, TRIANGLE], [SMALL, BLUE, TRIANGLE], 1.0]
[[RED, TRIANGLE], [RED, CIRCLE], 1.0]
[[RED, TRIANGLE], [MEDIUM, RED, CIRCLE], 1.0]
[[RED, TRIANGLE], [MEDIUM, RED, TRIANGLE], 2.0]
[[RED, TRIANGLE], [MEDIUM, BLUE, TRIANGLE], 1.0]
[[RED, TRIANGLE], [SMALL, TRIANGLE], 1.0]
[[RED, TRIANGLE], [SMALL, RED, TRIANGLE], 2.0]
[[RED, TRIANGLE], [SMALL, GREEN, TRIANGLE], 1.0]
[[RED, TRIANGLE], [SMALL, RED, SQUARE], 1.0]
[[RED, TRIANGLE], [RED, SQUARE], 1.0]
******************************************************
======================================================
>>>>> UnclassifiedResults <<<<<
======================================================
Total number of unclassified results = 24
======================================================
[[RED, TRIANGLE], [RED, TRIANGLE], 2.0]
[[RED, TRIANGLE], [BIG, RED, TRIANGLE], 2.0]
[[RED, TRIANGLE], [MEDIUM, RED, TRIANGLE], 2.0]
[[RED, TRIANGLE], [SMALL, RED, TRIANGLE], 2.0]
[[RED, TRIANGLE], [GREEN, TRIANGLE], 1.0]
```

```
[[RED, TRIANGLE], [MEDIUM, TRIANGLE], 1.0]
[[RED, TRIANGLE], [MEDIUM, GREEN, TRIANGLE], 1.0]
[[RED, TRIANGLE], [BIG, RED, CIRCLE], 1.0]
[[RED, TRIANGLE], [BIG, GREEN, TRIANGLE], 1.0]
[[RED, TRIANGLE], [MEDIUM, RED, SQUARE], 1.0]
[[RED, TRIANGLE], [SMALL, RED, CIRCLE], 1.0]
[[RED, TRIANGLE], [TRIANGLE], 1.0]
[[RED, TRIANGLE], [BIG, TRIANGLE], 1.0]
[[RED, TRIANGLE], [BIG, RED, SQUARE], 1.0]
[[RED, TRIANGLE], [SMALL, BLUE, TRIANGLE], 1.0]
[[RED, TRIANGLE], [RED, CIRCLE], 1.0]
[[RED, TRIANGLE], [MEDIUM, RED, CIRCLE], 1.0]
[[RED, TRIANGLE], [BLUE, TRIANGLE], 1.0]
[[RED, TRIANGLE], [MEDIUM, BLUE, TRIANGLE], 1.0]
[[RED, TRIANGLE], [SMALL, TRIANGLE], 1.0]
[[RED, TRIANGLE], [BIG, BLUE, TRIANGLE], 1.0]
[[RED, TRIANGLE], [SMALL, GREEN, TRIANGLE], 1.0]
[[RED, TRIANGLE], [SMALL, RED, SQUARE], 1.0]
[[RED, TRIANGLE], [RED, SQUARE], 1.0]
******************************************************
======================================================
>>>>> Classification Results <<<<<
======================================================
Total number of classified results = 24
======================================================
[[RED, TRIANGLE], [RED, TRIANGLE], 1]
[[RED, TRIANGLE], [BIG, RED, TRIANGLE], 1]
[[RED, TRIANGLE], [MEDIUM, RED, TRIANGLE], 1]
[[RED, TRIANGLE], [SMALL, RED, TRIANGLE], 1]
[[RED, TRIANGLE], [GREEN, TRIANGLE], 0]
[[RED, TRIANGLE], [MEDIUM, TRIANGLE], 0]
[[RED, TRIANGLE], [MEDIUM, GREEN, TRIANGLE], 0]
[[RED, TRIANGLE], [BIG, RED, CIRCLE], 0]
[[RED, TRIANGLE], [BIG, GREEN, TRIANGLE], 0]
[[RED, TRIANGLE], [MEDIUM, RED, SQUARE], 0]
[[RED, TRIANGLE], [SMALL, RED, CIRCLE], 0]
[[RED, TRIANGLE], [TRIANGLE], 0]
[[RED, TRIANGLE], [BIG, TRIANGLE], 0]
[[RED, TRIANGLE], [BIG, RED, SQUARE], 0]
[[RED, TRIANGLE], [SMALL, BLUE, TRIANGLE], 0]
[[RED, TRIANGLE], [RED, CIRCLE], 0]
[[RED, TRIANGLE], [MEDIUM, RED, CIRCLE], 0]
[[RED, TRIANGLE], [BLUE, TRIANGLE], 0]
[[RED, TRIANGLE], [MEDIUM, BLUE, TRIANGLE], 0]
[[RED, TRIANGLE], [SMALL, TRIANGLE], 0]
[[RED, TRIANGLE], [BIG, BLUE, TRIANGLE], 0]
[[RED, TRIANGLE], [SMALL, GREEN, TRIANGLE], 0]
[[RED, TRIANGLE], [SMALL, RED, SQUARE], 0]
[[RED, TRIANGLE], [RED, SQUARE], 0]
******************************************************
======================================================
>>>>>  NEW RULES <<<<<
======================================================
Total number of new Rules = 2
======================================================
```

```
[[SMALL, TRIANGLE], [RED, SMALL, TRIANGLE], 0.9]
[[RED, TRIANGLE], [RED, TRIANGLE], 3.375]
******************************************************
======================================================
>>>>>  DOCUMENTS FOR COMPARISON TO FOIL/WINNOW <<<<<
======================================================
Total number of documents in document file = 0
======================================================
******************************************************
======================================================
>>>>>  RULES BEFORE PROMOTION <<<<<
======================================================
Total number of Rules = 2
======================================================
[[SMALL, TRIANGLE], [RED, SMALL, TRIANGLE], 0.9]
[[RED, TRIANGLE], [RED, TRIANGLE], 3.375]
******************************************************
======================================================
>>>>>  RULES AFTER PROMOTION <<<<<
======================================================
Total number of Rules = 2
======================================================
[[SMALL, TRIANGLE], [RED, SMALL, TRIANGLE], 0.9]
[[RED, TRIANGLE], [RED, TRIANGLE], 3.375]
******************************************************
======================================================
This file contains the comparison data for testing.
======================================================
>>>>> Search Results <<<<<
======================================================
Total number of search results = 24
======================================================
[[RED, TRIANGLE], [BLUE, TRIANGLE], 1.0]
[[RED, TRIANGLE], [BIG, GREEN, TRIANGLE], 1.0]
[[RED, TRIANGLE], [RED, TRIANGLE], 20.140625]
[[RED, TRIANGLE], [BIG, BLUE, TRIANGLE], 1.0]
[[RED, TRIANGLE], [GREEN, TRIANGLE], 1.0]
[[RED, TRIANGLE], [MEDIUM, TRIANGLE], 1.0]
[[RED, TRIANGLE], [MEDIUM, GREEN, TRIANGLE], 1.0]
[[RED, TRIANGLE], [BIG, RED, CIRCLE], 1.0]
[[RED, TRIANGLE], [BIG, RED, TRIANGLE], 20.140625]
[[RED, TRIANGLE], [MEDIUM, RED, SQUARE], 1.0]
[[RED, TRIANGLE], [SMALL, RED, CIRCLE], 1.0]
[[RED, TRIANGLE], [TRIANGLE], 1.0]
[[RED, TRIANGLE], [BIG, TRIANGLE], 1.0]
[[RED, TRIANGLE], [BIG, RED, SQUARE], 1.0]
[[RED, TRIANGLE], [SMALL, BLUE, TRIANGLE], 1.0]
[[RED, TRIANGLE], [RED, CIRCLE], 1.0]
[[RED, TRIANGLE], [MEDIUM, RED, CIRCLE], 1.0]
[[RED, TRIANGLE], [MEDIUM, RED, TRIANGLE], 20.140625]
[[RED, TRIANGLE], [MEDIUM, BLUE, TRIANGLE], 1.0]
[[RED, TRIANGLE], [SMALL, TRIANGLE], 1.0]
[[RED, TRIANGLE], [SMALL, RED, TRIANGLE], 20.140625]
[[RED, TRIANGLE], [SMALL, GREEN, TRIANGLE], 1.0]
[[RED, TRIANGLE], [SMALL, RED, SQUARE], 1.0]
```

```
[[RED, TRIANGLE], [RED, SQUARE], 1.0]
******************************************************
======================================================
>>>>> UnclassifiedResults <<<<<
======================================================
Total number of unclassified results = 24
======================================================
[[RED, TRIANGLE], [RED, TRIANGLE], 20.140625]
[[RED, TRIANGLE], [BIG, RED, TRIANGLE], 20.140625]
[[RED, TRIANGLE], [MEDIUM, RED, TRIANGLE], 20.140625]
[[RED, TRIANGLE], [SMALL, RED, TRIANGLE], 20.140625]
[[RED, TRIANGLE], [GREEN, TRIANGLE], 1.0]
[[RED, TRIANGLE], [MEDIUM, TRIANGLE], 1.0]
[[RED, TRIANGLE], [MEDIUM, GREEN, TRIANGLE], 1.0]
[[RED, TRIANGLE], [BIG, RED, CIRCLE], 1.0]
[[RED, TRIANGLE], [BIG, GREEN, TRIANGLE], 1.0]
[[RED, TRIANGLE], [MEDIUM, RED, SQUARE], 1.0]
[[RED, TRIANGLE], [SMALL, RED, CIRCLE], 1.0]
[[RED, TRIANGLE], [TRIANGLE], 1.0]
[[RED, TRIANGLE], [BIG, TRIANGLE], 1.0]
[[RED, TRIANGLE], [BIG, RED, SQUARE], 1.0]
[[RED, TRIANGLE], [SMALL, BLUE, TRIANGLE], 1.0]
[[RED, TRIANGLE], [RED, CIRCLE], 1.0]
[[RED, TRIANGLE], [MEDIUM, RED, CIRCLE], 1.0]
[[RED, TRIANGLE], [BLUE, TRIANGLE], 1.0]
[[RED, TRIANGLE], [MEDIUM, BLUE, TRIANGLE], 1.0]
[[RED, TRIANGLE], [SMALL, TRIANGLE], 1.0]
[[RED, TRIANGLE], [BIG, BLUE, TRIANGLE], 1.0]
[[RED, TRIANGLE], [SMALL, GREEN, TRIANGLE], 1.0]
[[RED, TRIANGLE], [SMALL, RED, SQUARE], 1.0]
[[RED, TRIANGLE], [RED, SQUARE], 1.0]
******************************************************
======================================================
>>>>> Classification Results <<<<<
======================================================
Total number of classified results = 24
======================================================
[[RED, TRIANGLE], [RED, TRIANGLE], 0]
[[RED, TRIANGLE], [BIG, RED, TRIANGLE], 1]
[[RED, TRIANGLE], [MEDIUM, RED, TRIANGLE], 0]
[[RED, TRIANGLE], [SMALL, RED, TRIANGLE], 0]
[[RED, TRIANGLE], [GREEN, TRIANGLE], 0]
[[RED, TRIANGLE], [MEDIUM, TRIANGLE], 0]
[[RED, TRIANGLE], [MEDIUM, GREEN, TRIANGLE], 0]
[[RED, TRIANGLE], [BIG, RED, CIRCLE], 0]
[[RED, TRIANGLE], [BIG, GREEN, TRIANGLE], 0]
[[RED, TRIANGLE], [MEDIUM, RED, SQUARE], 0]
[[RED, TRIANGLE], [SMALL, RED, CIRCLE], 0]
[[RED, TRIANGLE], [TRIANGLE], 0]
[[RED, TRIANGLE], [BIG, TRIANGLE], 0]
[[RED, TRIANGLE], [BIG, RED, SQUARE], 0]
[[RED, TRIANGLE], [SMALL, BLUE, TRIANGLE], 0]
[[RED, TRIANGLE], [RED, CIRCLE], 0]
[[RED, TRIANGLE], [MEDIUM, RED, CIRCLE], 0]
[[RED, TRIANGLE], [BLUE, TRIANGLE], 0]
```

```
[[RED, TRIANGLE], [MEDIUM, BLUE, TRIANGLE], 0]
[[RED, TRIANGLE], [SMALL, TRIANGLE], 0]
[[RED, TRIANGLE], [BIG, BLUE, TRIANGLE], 0]
[[RED, TRIANGLE], [SMALL, GREEN, TRIANGLE], 0]
[[RED, TRIANGLE], [SMALL, RED, SQUARE], 0]
[[RED, TRIANGLE], [RED, SQUARE], 0]
******************************************************
======================================================
>>>>>  NEW RULES <<<<<
======================================================
Total number of new Rules = 3
======================================================
[[SMALL, TRIANGLE], [RED, SMALL, TRIANGLE], 0.81]
[[RED, TRIANGLE], [RED, TRIANGLE], 3.690562]
[[RED, TRIANGLE], [BIG, RED, TRIANGLE], 1.0]
******************************************************
======================================================
>>>>>  DOCUMENTS FOR COMPARISON TO FOIL/WINNOW <<<<<
======================================================
Total number of documents in document file = 0
======================================================
******************************************************
======================================================
>>>>>  RULES BEFORE PROMOTION <<<<<
======================================================
Total number of Rules = 3
======================================================
[[SMALL, TRIANGLE], [RED, SMALL, TRIANGLE], 0.81]
[[RED, TRIANGLE], [RED, TRIANGLE], 3.690562]
[[RED, TRIANGLE], [BIG, RED, TRIANGLE], 1.0]
******************************************************
======================================================
>>>>>  RULES AFTER PROMOTION <<<<<
======================================================
Total number of Rules = 3
======================================================
[[SMALL, TRIANGLE], [RED, SMALL, TRIANGLE], 0.81]
[[RED, TRIANGLE], [RED, TRIANGLE], 3.690562]
[[RED, TRIANGLE], [BIG, RED, TRIANGLE], 1.0]
******************************************************
======================================================
This file contains the comparison data for testing.
======================================================
>>>>> Search Results <<<<<
======================================================
Total number of search results = 24
======================================================
[[RED, TRIANGLE], [BLUE, TRIANGLE], 1.0]
[[RED, TRIANGLE], [BIG, GREEN, TRIANGLE], 1.0]
[[RED, TRIANGLE], [RED, TRIANGLE], 23.001371]
[[RED, TRIANGLE], [BIG, BLUE, TRIANGLE], 1.0]
[[RED, TRIANGLE], [GREEN, TRIANGLE], 1.0]
[[RED, TRIANGLE], [MEDIUM, TRIANGLE], 1.0]
[[RED, TRIANGLE], [MEDIUM, GREEN, TRIANGLE], 1.0]
[[RED, TRIANGLE], [BIG, RED, CIRCLE], 1.0]
```

```
[[RED, TRIANGLE], [BIG, RED, TRIANGLE], 26.001371]
[[RED, TRIANGLE], [MEDIUM, RED, SQUARE], 1.0]
[[RED, TRIANGLE], [SMALL, RED, CIRCLE], 1.0]
[[RED, TRIANGLE], [TRIANGLE], 1.0]
[[RED, TRIANGLE], [BIG, TRIANGLE], 1.0]
[[RED, TRIANGLE], [BIG, RED, SQUARE], 1.0]
[[RED, TRIANGLE], [SMALL, BLUE, TRIANGLE], 1.0]
[[RED, TRIANGLE], [RED, CIRCLE], 1.0]
[[RED, TRIANGLE], [MEDIUM, RED, CIRCLE], 1.0]
[[RED, TRIANGLE], [MEDIUM, RED, TRIANGLE], 23.001371]
[[RED, TRIANGLE], [MEDIUM, BLUE, TRIANGLE], 1.0]
[[RED, TRIANGLE], [SMALL, TRIANGLE], 1.0]
[[RED, TRIANGLE], [SMALL, RED, TRIANGLE], 23.001371]
[[RED, TRIANGLE], [SMALL, GREEN, TRIANGLE], 1.0]
[[RED, TRIANGLE], [SMALL, RED, SQUARE], 1.0]
[[RED, TRIANGLE], [RED, SQUARE], 1.0]
******************************************************
======================================================
>>>>> UnclassifiedResults <<<<<
======================================================
Total number of unclassified results = 24
======================================================

[[RED, TRIANGLE], [BIG, RED, TRIANGLE], 26.001371]
[[RED, TRIANGLE], [RED, TRIANGLE], 23.001371]
[[RED, TRIANGLE], [MEDIUM, RED, TRIANGLE], 23.001371]
[[RED, TRIANGLE], [SMALL, RED, TRIANGLE], 23.001371]
[[RED, TRIANGLE], [GREEN, TRIANGLE], 1.0]
[[RED, TRIANGLE], [MEDIUM, TRIANGLE], 1.0]
[[RED, TRIANGLE], [MEDIUM, GREEN, TRIANGLE], 1.0]
[[RED, TRIANGLE], [BIG, RED, CIRCLE], 1.0]
[[RED, TRIANGLE], [BIG, GREEN, TRIANGLE], 1.0]
[[RED, TRIANGLE], [MEDIUM, RED, SQUARE], 1.0]
[[RED, TRIANGLE], [SMALL, RED, CIRCLE], 1.0]
[[RED, TRIANGLE], [TRIANGLE], 1.0]
[[RED, TRIANGLE], [BIG, TRIANGLE], 1.0]
[[RED, TRIANGLE], [BIG, RED, SQUARE], 1.0]
[[RED, TRIANGLE], [SMALL, BLUE, TRIANGLE], 1.0]
[[RED, TRIANGLE], [RED, CIRCLE], 1.0]
[[RED, TRIANGLE], [MEDIUM, RED, CIRCLE], 1.0]
[[RED, TRIANGLE], [BLUE, TRIANGLE], 1.0]
[[RED, TRIANGLE], [MEDIUM, BLUE, TRIANGLE], 1.0]
[[RED, TRIANGLE], [SMALL, TRIANGLE], 1.0]
[[RED, TRIANGLE], [BIG, BLUE, TRIANGLE], 1.0]
[[RED, TRIANGLE], [SMALL, GREEN, TRIANGLE], 1.0]
[[RED, TRIANGLE], [SMALL, RED, SQUARE], 1.0]
[[RED, TRIANGLE], [RED, SQUARE], 1.0]
******************************************************
======================================================
>>>>> Classification Results <<<<<
======================================================
Total number of classified results = 24
======================================================
[[RED, TRIANGLE], [BIG, RED, TRIANGLE], 0]
[[RED, TRIANGLE], [RED, TRIANGLE], 0]
```

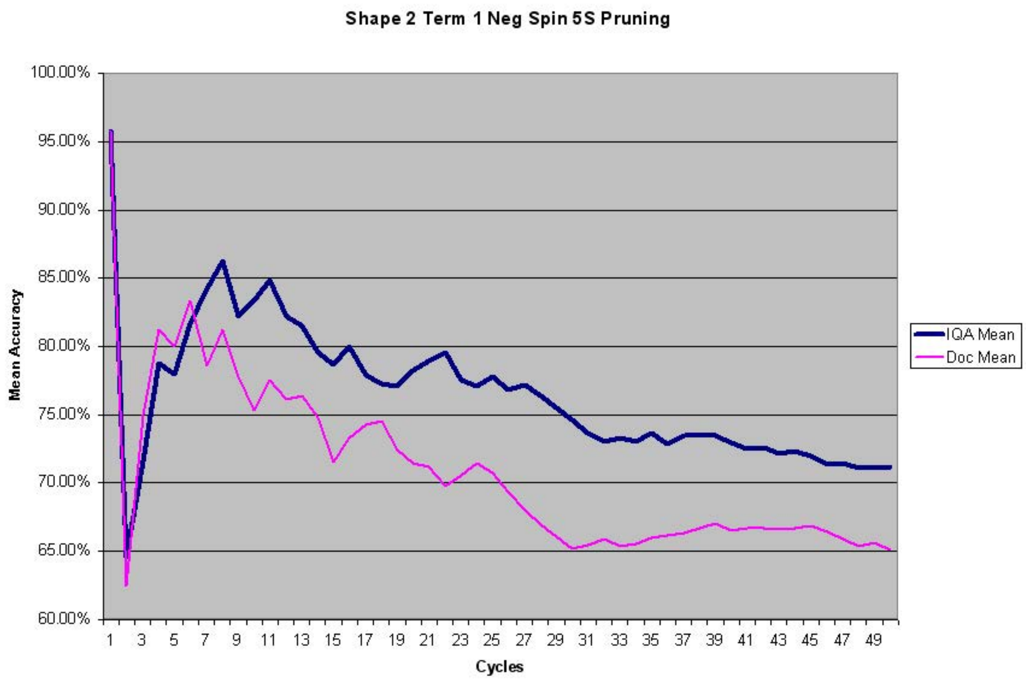
```
[[RED, TRIANGLE], [MEDIUM, RED, TRIANGLE], 1]
[[RED, TRIANGLE], [SMALL, RED, TRIANGLE], 0]
[[RED, TRIANGLE], [GREEN, TRIANGLE], 0]
[[RED, TRIANGLE], [MEDIUM, TRIANGLE], 0]
[[RED, TRIANGLE], [MEDIUM, GREEN, TRIANGLE], 0]
[[RED, TRIANGLE], [BIG, RED, CIRCLE], 0]
[[RED, TRIANGLE], [BIG, GREEN, TRIANGLE], 0]
[[RED, TRIANGLE], [MEDIUM, RED, SQUARE], 0]
[[RED, TRIANGLE], [SMALL, RED, CIRCLE], 0]
[[RED, TRIANGLE], [TRIANGLE], 0]
[[RED, TRIANGLE], [BIG, TRIANGLE], 0]
[[RED, TRIANGLE], [BIG, RED, SQUARE], 0]
[[RED, TRIANGLE], [SMALL, BLUE, TRIANGLE], 0]
[[RED, TRIANGLE], [RED, CIRCLE], 0]
[[RED, TRIANGLE], [MEDIUM, RED, CIRCLE], 0]
[[RED, TRIANGLE], [BLUE, TRIANGLE], 0]
[[RED, TRIANGLE], [MEDIUM, BLUE, TRIANGLE], 0]
[[RED, TRIANGLE], [SMALL, TRIANGLE], 0]
[[RED, TRIANGLE], [BIG, BLUE, TRIANGLE], 0]
[[RED, TRIANGLE], [SMALL, GREEN, TRIANGLE], 0]
[[RED, TRIANGLE], [SMALL, RED, SQUARE], 0]
[[RED, TRIANGLE], [RED, SQUARE], 0]
******************************************************
======================================================
>>>>>  NEW RULES <<<<<
======================================================
Total number of new Rules = 4
======================================================
[[SMALL, TRIANGLE], [RED, SMALL, TRIANGLE], 0.729]
[[RED, TRIANGLE], [RED, TRIANGLE], 4.0356293]
[[RED, TRIANGLE], [BIG, RED, TRIANGLE], 0.9]
[[RED, TRIANGLE], [MEDIUM, RED, TRIANGLE], 1.0]
******************************************************
======================================================
>>>>>  DOCUMENTS FOR COMPARISON TO FOIL/WINNOW <<<<<
======================================================
Total number of documents in document file = 0
======================================================
******************************************************
======================================================
>>>>>  RULES BEFORE PROMOTION <<<<<
======================================================
Total number of Rules = 4
======================================================
[[SMALL, TRIANGLE], [RED, SMALL, TRIANGLE], 0.729]
[[RED, TRIANGLE], [RED, TRIANGLE], 4.0356293]
[[RED, TRIANGLE], [BIG, RED, TRIANGLE], 0.9]
[[RED, TRIANGLE], [MEDIUM, RED, TRIANGLE], 1.0]
******************************************************
======================================================
>>>>>  RULES AFTER PROMOTION <<<<<
======================================================
Total number of Rules = 4
======================================================
[[SMALL, TRIANGLE], [RED, SMALL, TRIANGLE], 0.729]
```

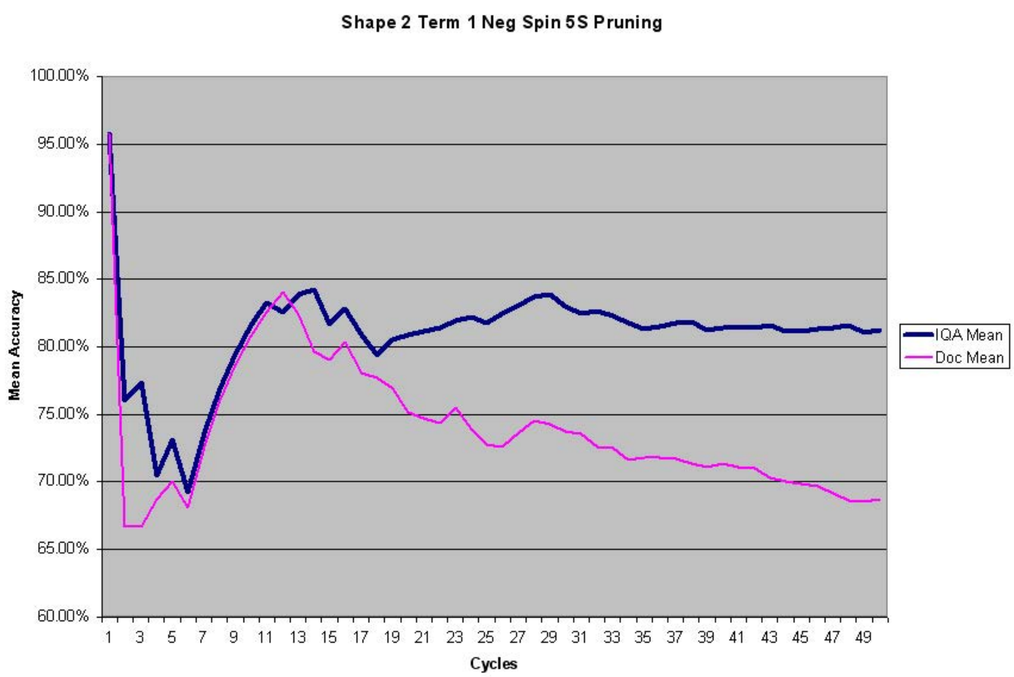
```
[[RED, TRIANGLE], [RED, TRIANGLE], 4.0356293]
[[RED, TRIANGLE], [BIG, RED, TRIANGLE], 0.9]
[[RED, TRIANGLE], [MEDIUM, RED, TRIANGLE], 1.0]
******************************************************
======================================================
This file contains the comparison data for testing.
======================================================
>>>>> Search Results <<<<<
======================================================
Total number of search results = 24
======================================================
[[RED, TRIANGLE], [BLUE, TRIANGLE], 1.0]
[[RED, TRIANGLE], [BIG, GREEN, TRIANGLE], 1.0]
[[RED, TRIANGLE], [RED, TRIANGLE], 26.357563]
[[RED, TRIANGLE], [BIG, BLUE, TRIANGLE], 1.0]
[[RED, TRIANGLE], [GREEN, TRIANGLE], 1.0]
[[RED, TRIANGLE], [MEDIUM, TRIANGLE], 1.0]
[[RED, TRIANGLE], [MEDIUM, GREEN, TRIANGLE], 1.0]
[[RED, TRIANGLE], [BIG, RED, CIRCLE], 1.0]
[[RED, TRIANGLE], [BIG, RED, TRIANGLE], 26.531805]
[[RED, TRIANGLE], [MEDIUM, RED, SQUARE], 1.0]
[[RED, TRIANGLE], [SMALL, RED, CIRCLE], 1.0]
[[RED, TRIANGLE], [TRIANGLE], 1.0]
[[RED, TRIANGLE], [BIG, TRIANGLE], 1.0]
[[RED, TRIANGLE], [BIG, RED, SQUARE], 1.0]
[[RED, TRIANGLE], [SMALL, BLUE, TRIANGLE], 1.0]
[[RED, TRIANGLE], [RED, CIRCLE], 1.0]
[[RED, TRIANGLE], [MEDIUM, RED, CIRCLE], 1.0]
[[RED, TRIANGLE], [MEDIUM, RED, TRIANGLE], 29.357563]
[[RED, TRIANGLE], [MEDIUM, BLUE, TRIANGLE], 1.0]
[[RED, TRIANGLE], [SMALL, TRIANGLE], 1.0]
[[RED, TRIANGLE], [SMALL, RED, TRIANGLE], 26.357563]
[[RED, TRIANGLE], [SMALL, GREEN, TRIANGLE], 1.0]
[[RED, TRIANGLE], [SMALL, RED, SQUARE], 1.0]
[[RED, TRIANGLE], [RED, SQUARE], 1.0]
******************************************************
======================================================
>>>>> UnclassifiedResults <<<<<
======================================================
Total number of unclassified results = 24
======================================================
[[RED, TRIANGLE], [MEDIUM, RED, TRIANGLE], 29.357563]
[[RED, TRIANGLE], [BIG, RED, TRIANGLE], 26.531805]
[[RED, TRIANGLE], [RED, TRIANGLE], 26.357563]
[[RED, TRIANGLE], [SMALL, RED, TRIANGLE], 26.357563]
[[RED, TRIANGLE], [GREEN, TRIANGLE], 1.0]
[[RED, TRIANGLE], [MEDIUM, TRIANGLE], 1.0]
[[RED, TRIANGLE], [MEDIUM, GREEN, TRIANGLE], 1.0]
[[RED, TRIANGLE], [BIG, RED, CIRCLE], 1.0]
[[RED, TRIANGLE], [BIG, GREEN, TRIANGLE], 1.0]
[[RED, TRIANGLE], [MEDIUM, RED, SQUARE], 1.0]
[[RED, TRIANGLE], [SMALL, RED, CIRCLE], 1.0]
[[RED, TRIANGLE], [TRIANGLE], 1.0]
[[RED, TRIANGLE], [BIG, TRIANGLE], 1.0]
[[RED, TRIANGLE], [BIG, RED, SQUARE], 1.0]
```

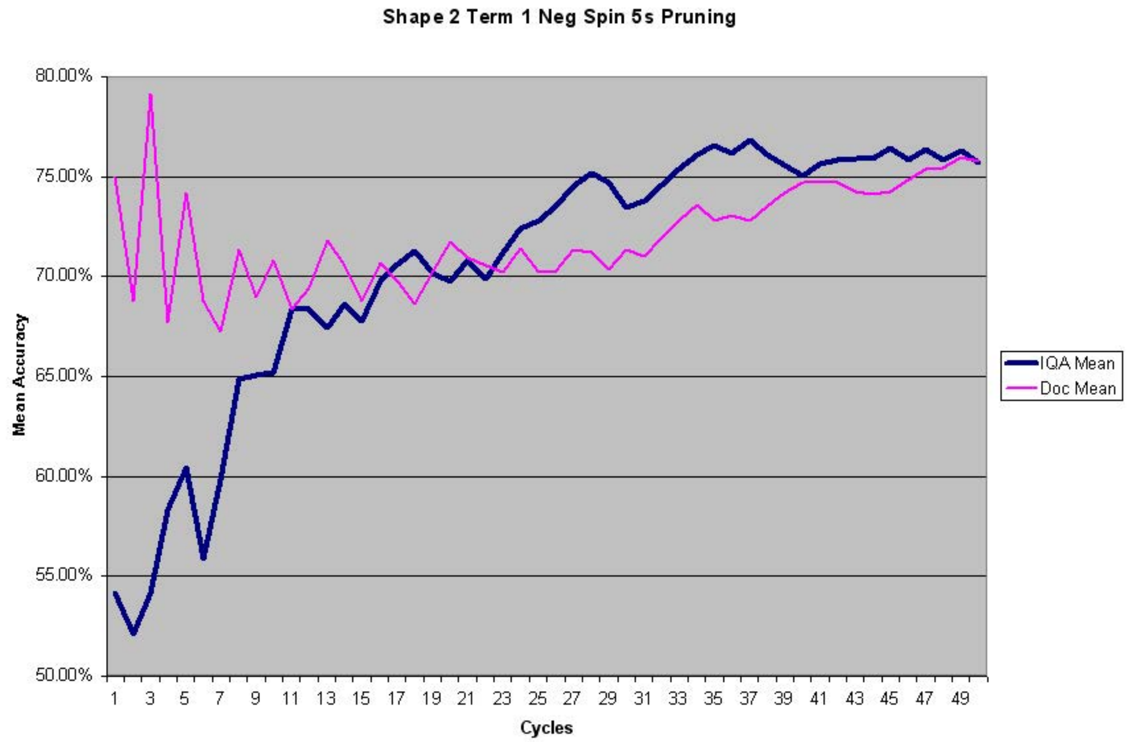
```
[[RED, TRIANGLE], [SMALL, BLUE, TRIANGLE], 1.0]
[[RED, TRIANGLE], [RED, CIRCLE], 1.0]
[[RED, TRIANGLE], [MEDIUM, RED, CIRCLE], 1.0]
[[RED, TRIANGLE], [BLUE, TRIANGLE], 1.0]
[[RED, TRIANGLE], [MEDIUM, BLUE, TRIANGLE], 1.0]
[[RED, TRIANGLE], [SMALL, TRIANGLE], 1.0]
[[RED, TRIANGLE], [BIG, BLUE, TRIANGLE], 1.0]
[[RED, TRIANGLE], [SMALL, GREEN, TRIANGLE], 1.0]
[[RED, TRIANGLE], [SMALL, RED, SQUARE], 1.0]
[[RED, TRIANGLE], [RED, SQUARE], 1.0]
******************************************************
```

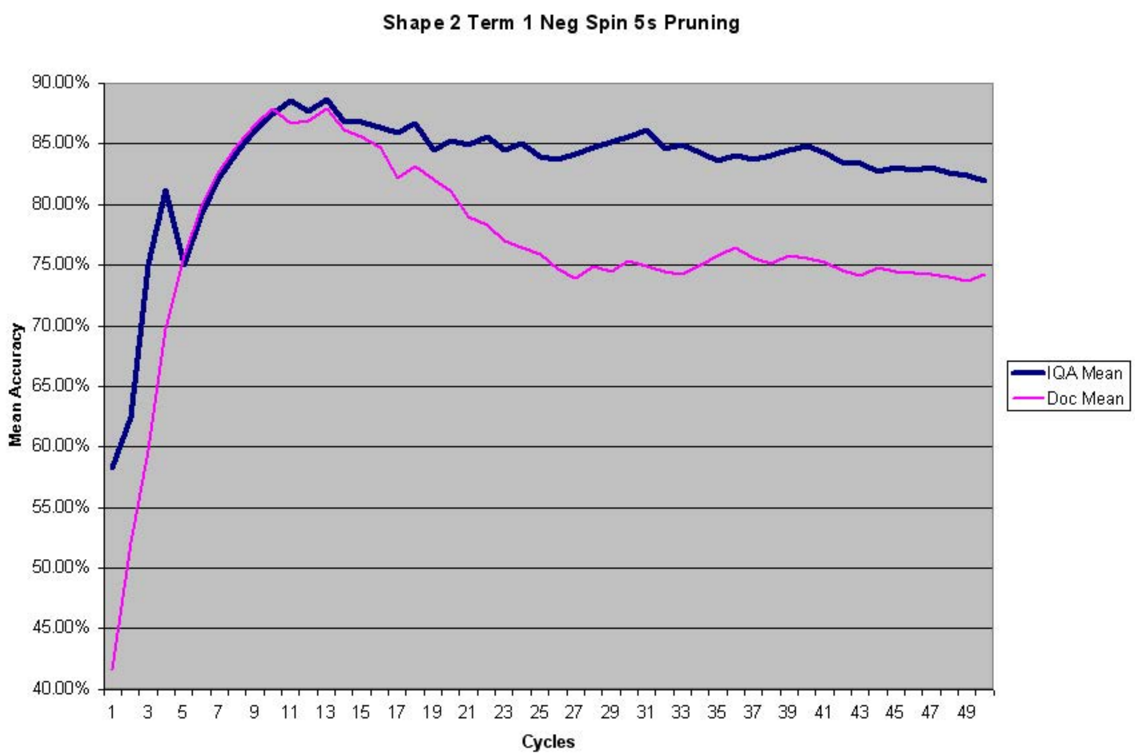
Appendix D – Test Graphs

1. 'Shape Dataset' 2-Term-1





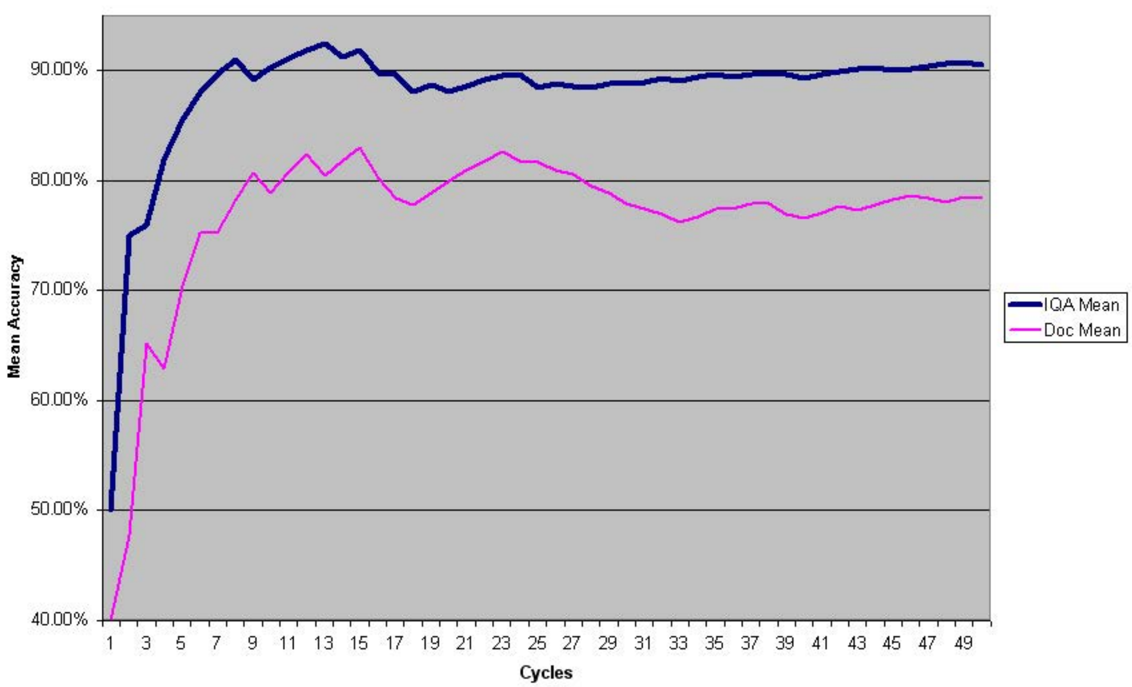

Shape 2 Term 1 Neg Spin 5s Pruning
80.00%
75.00%
70.00%
65.00%
60.00%
55.00%
50.00%
Mean Accuracy
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31 33 35 37 39 41 43 45 47 49
Cycles
IQA Mean
Doc Mean


Shape 2 Term 1 Neg Spin 5s Pruning
90.00%
85.00%
80.00%
75.00%
70.00%
65.00%
60.00%
55.00%
50.00%
45.00%
40.00%
Mean Accuracy
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31 33 35 37 39 41 43 45 47 49
Cycles
IQA Mean
Doc Mean

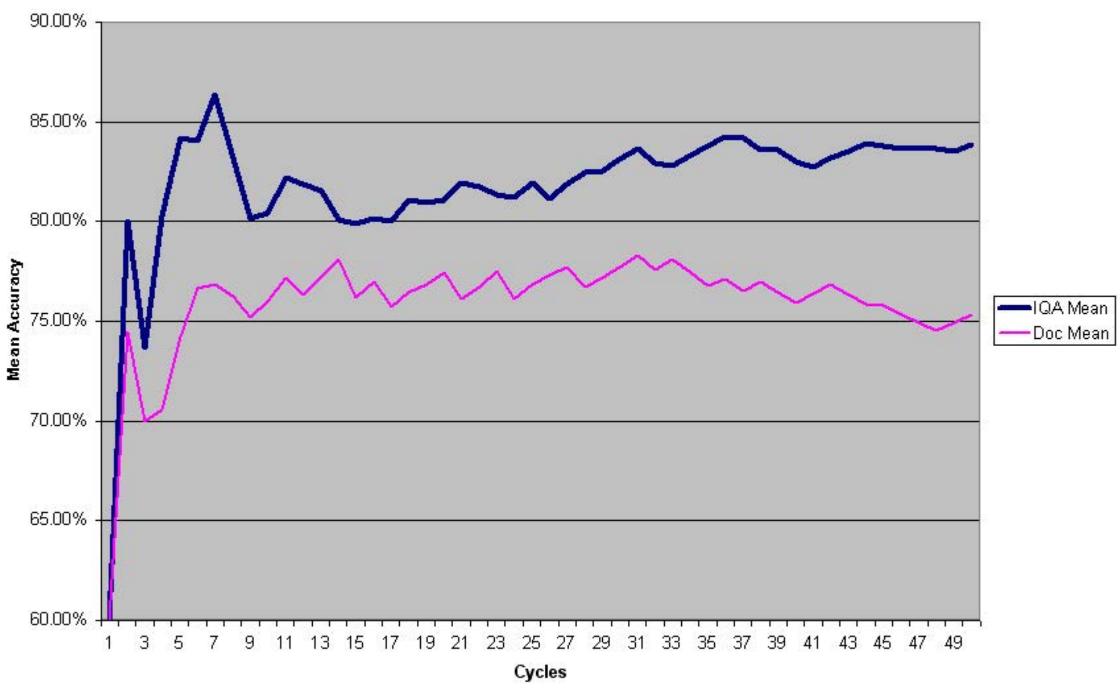

Shape 2 Term 1 Neg Spin 5s Pruning
80.00%
75.00%
70.00%
65.00%
60.00%
55.00%
50.00%
45.00%
40.00%
35.00%
30.00%
Mean Accuracy
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31 33 35 37 39 41 43 45 47 49
Cycles
IQA Mean
Doc Mean

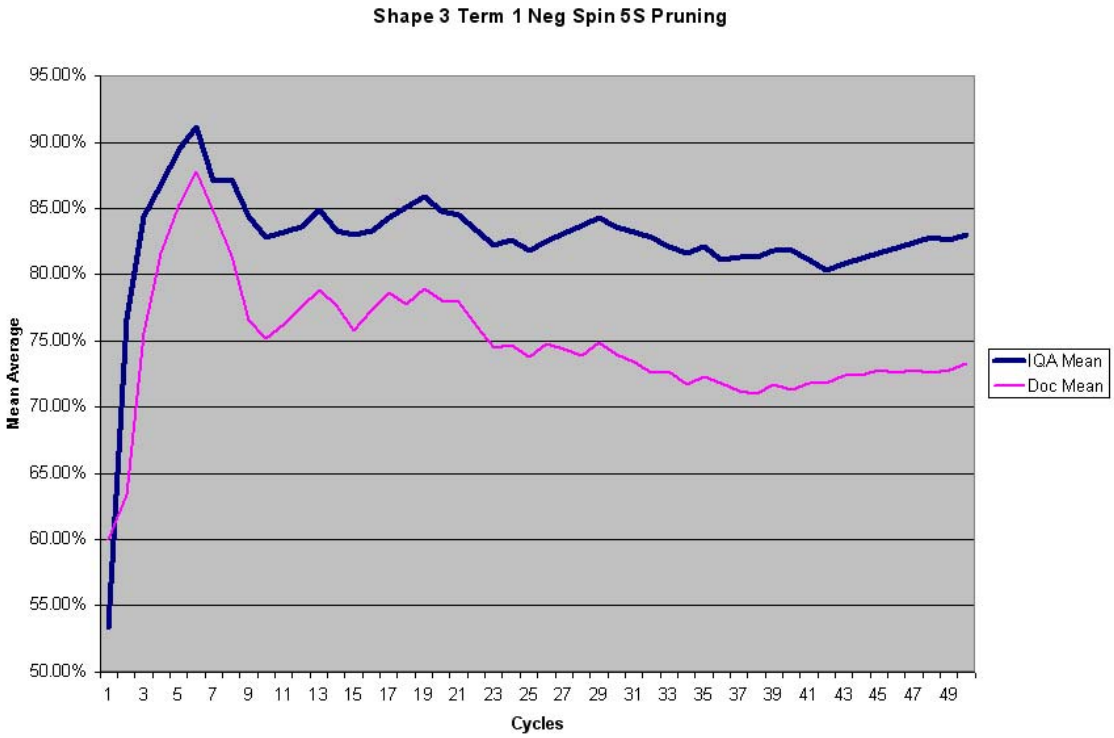
## 2.	‘Shape Dataset’ 3-Term-1

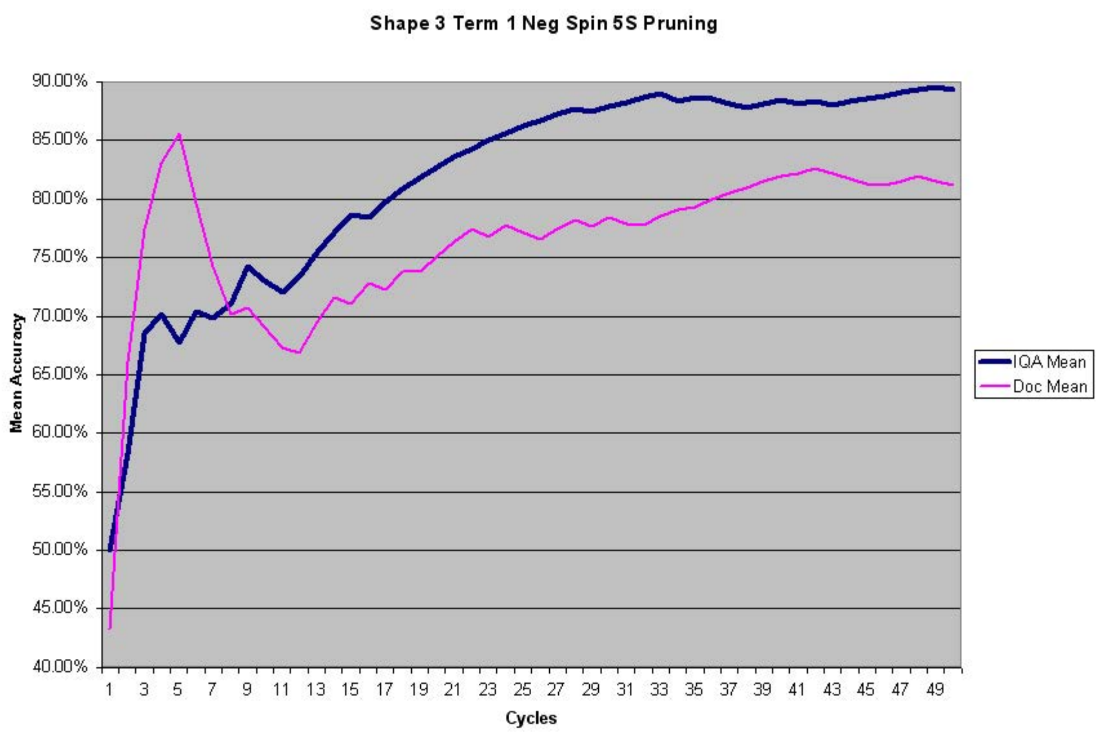
Shape 3 Term 2 Neg Spin 5S Pruning

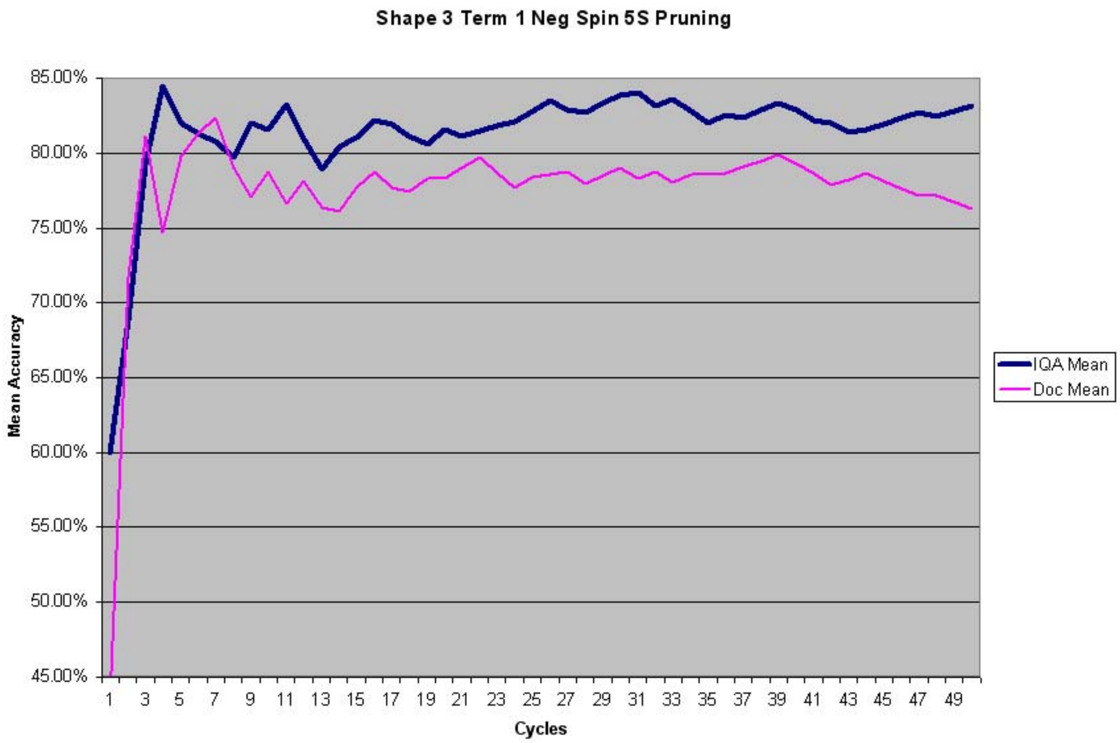


Shape 3 Term 1 Neg Spin Pruning

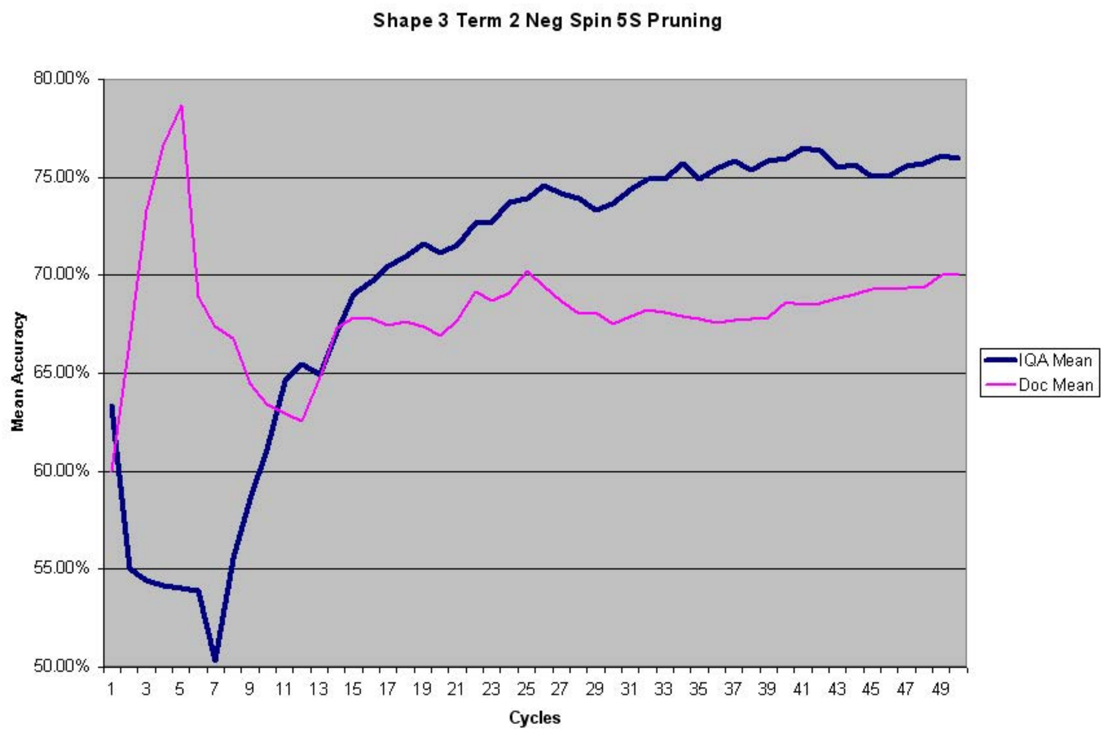


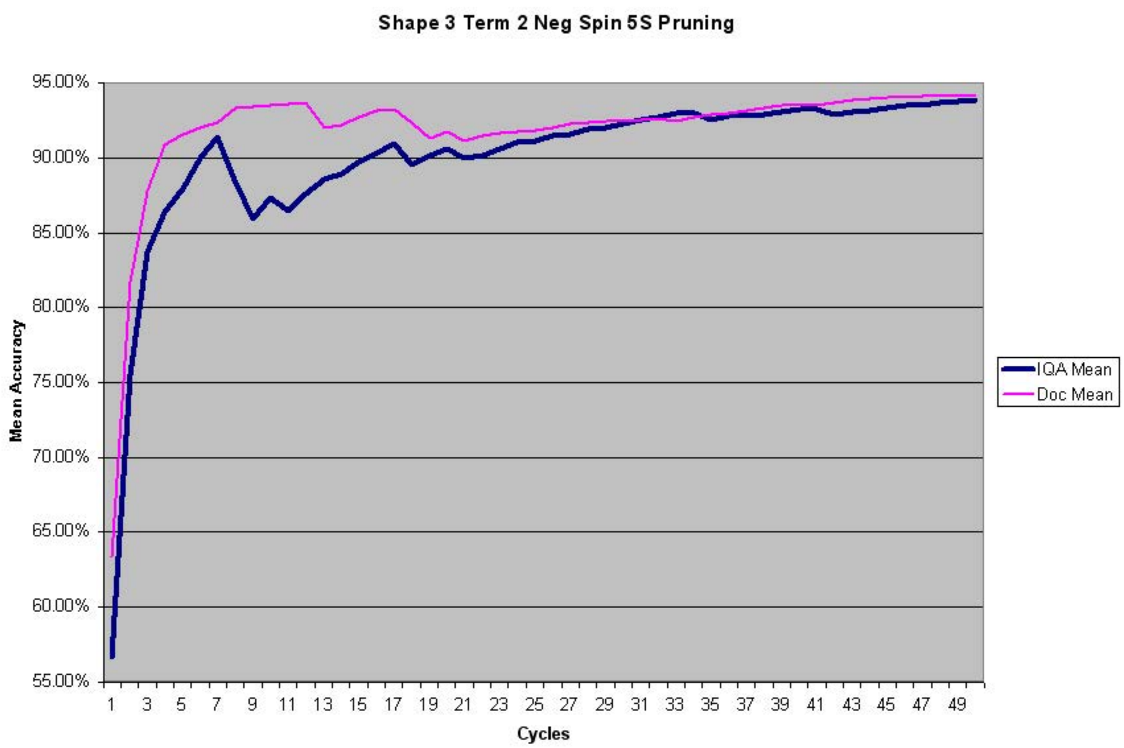

Shape 3 Term 1 Neg Spin 5S Pruning
Mean Average
Cycles
IQA Mean
Doc Mean


Shape 3 Term 1 Neg Spin 5S Pruning
Mean Accuracy
Cycles
IQA Mean
Doc Mean

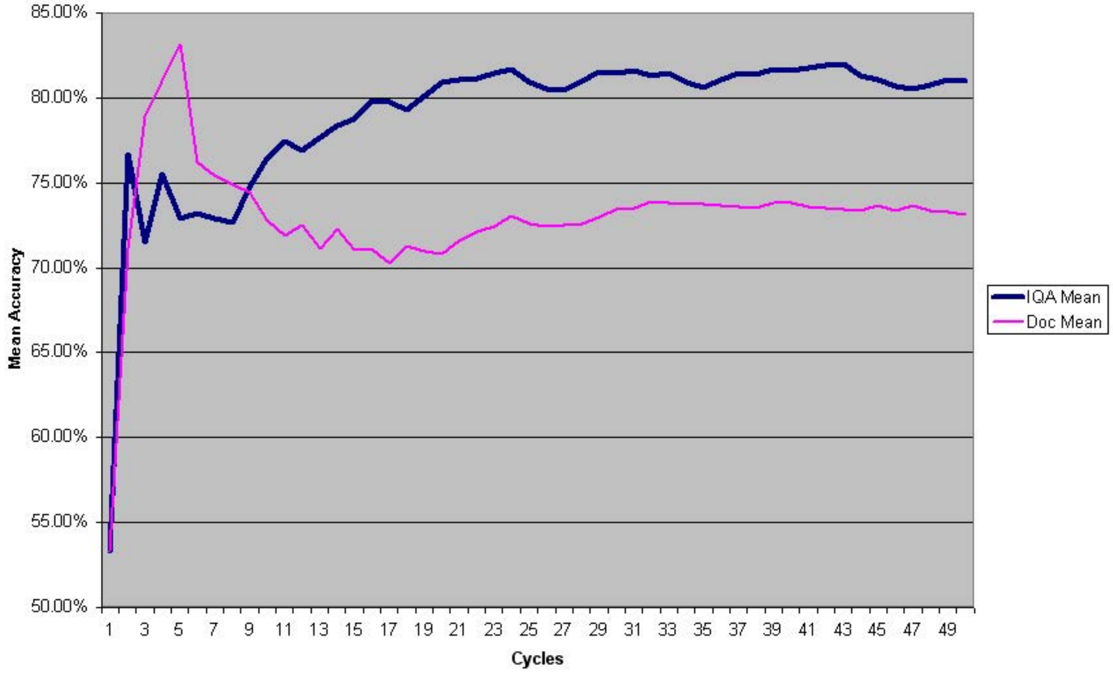

Shape 3 Term 1 Neg Spin 5S Pruning
85.00%
80.00%
75.00%
70.00%
65.00%
60.00%
55.00%
50.00%
45.00%
Mean Accuracy
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31 33 35 37 39 41 43 45 47 49
Cycles
IQA Mean
Doc Mean

3. 'Shape Dataset' 3-Term-2

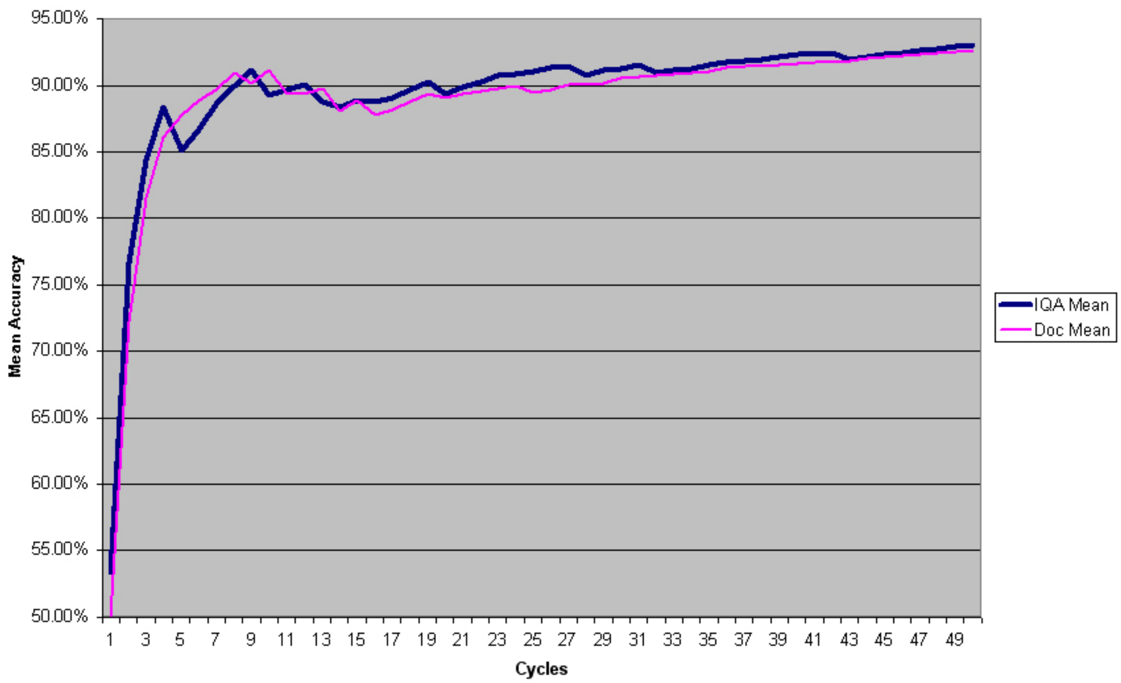


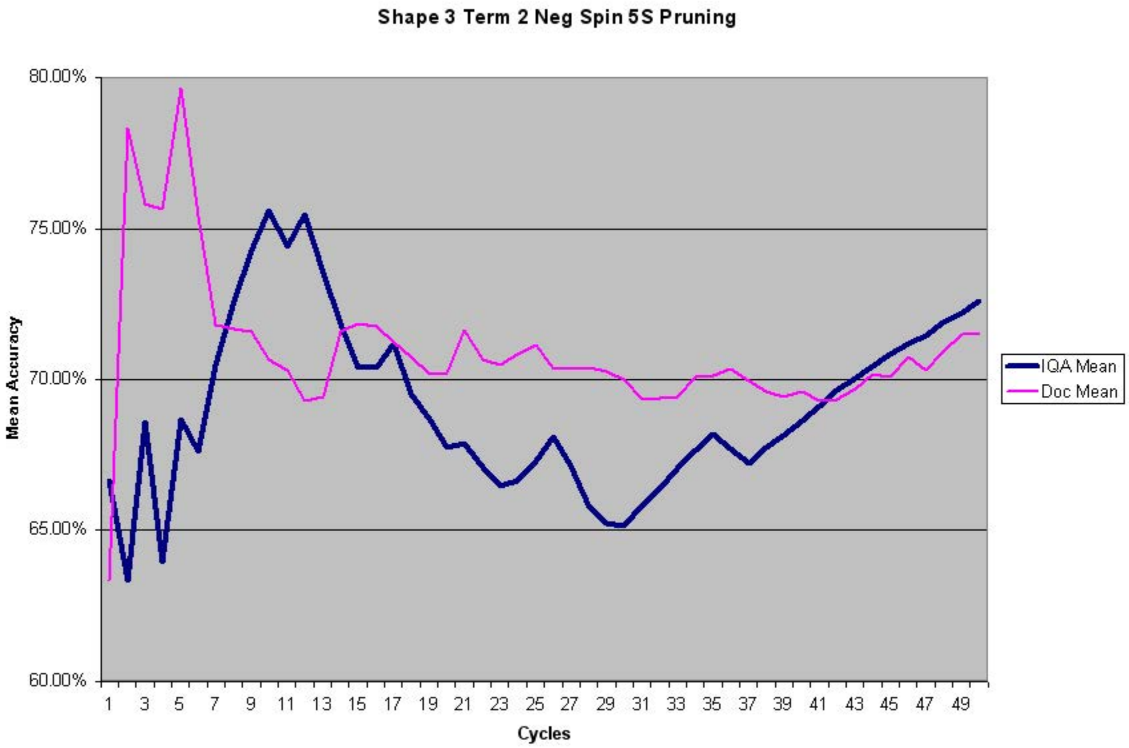


Shape 3 Term 2 Neg Spin 5S Pruning

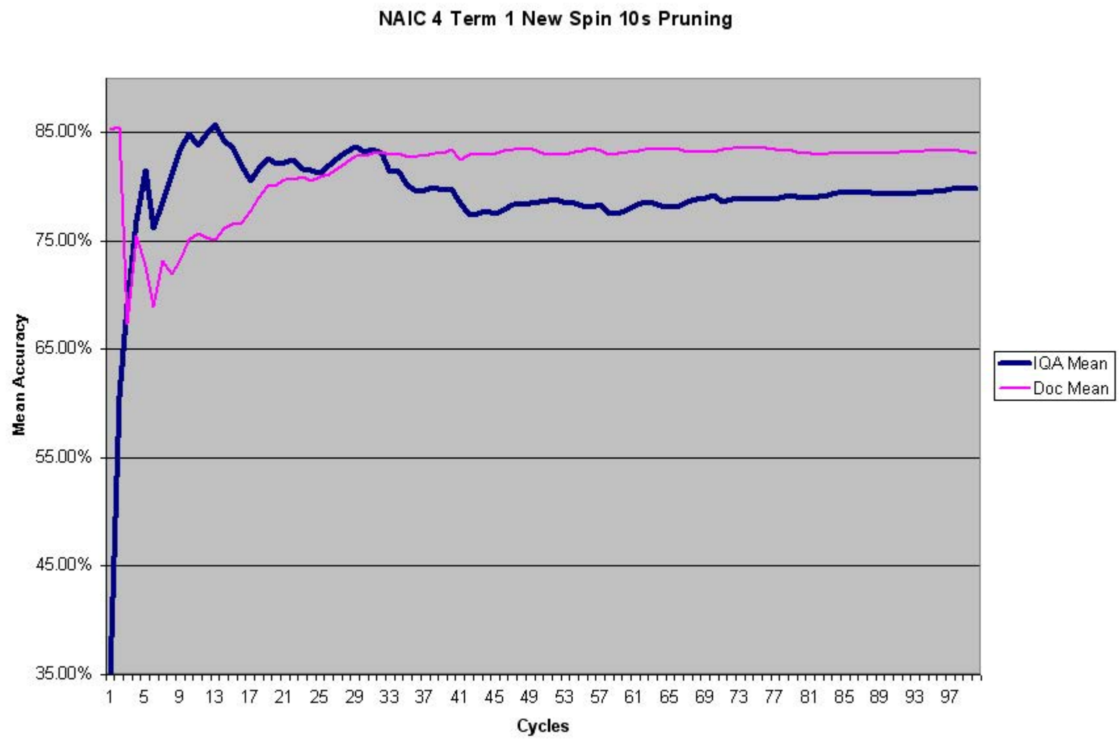


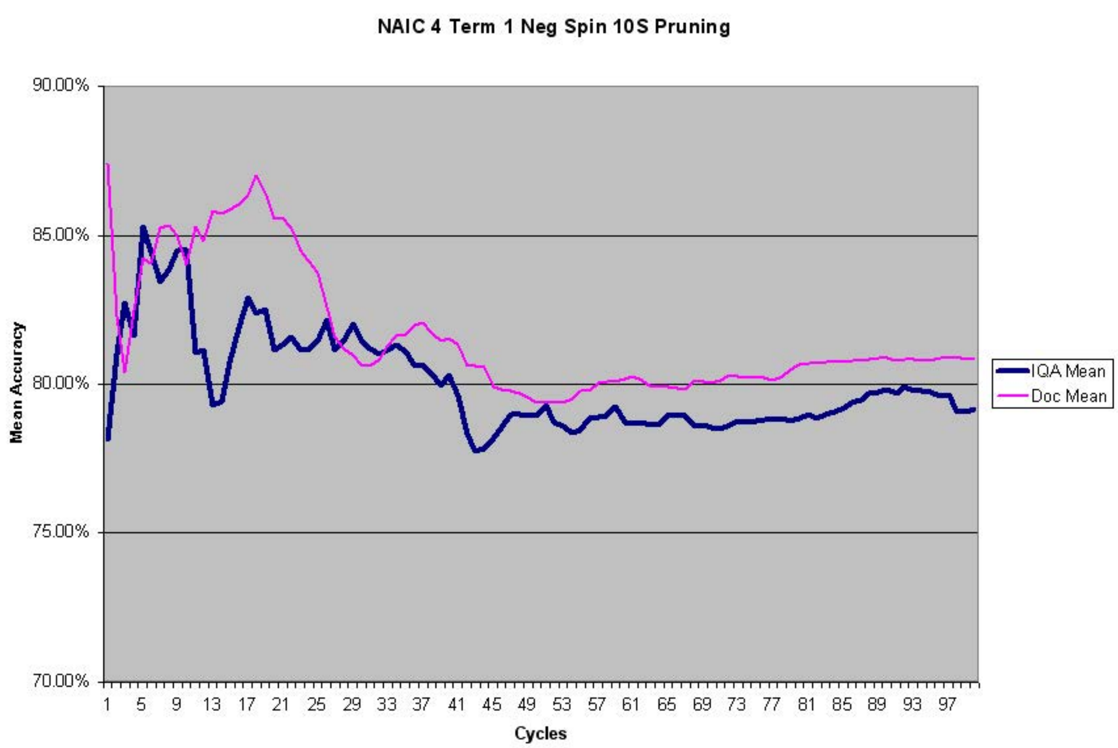
Shape 3 Term 2 Neg Spin 5S Pruning

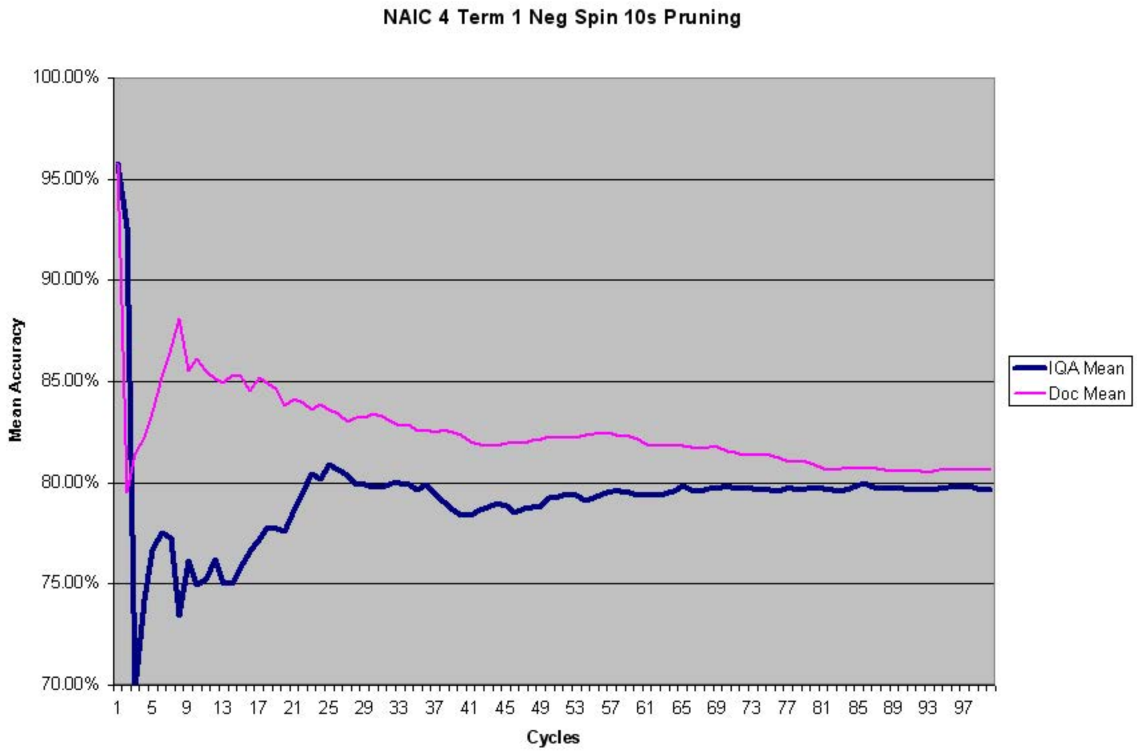


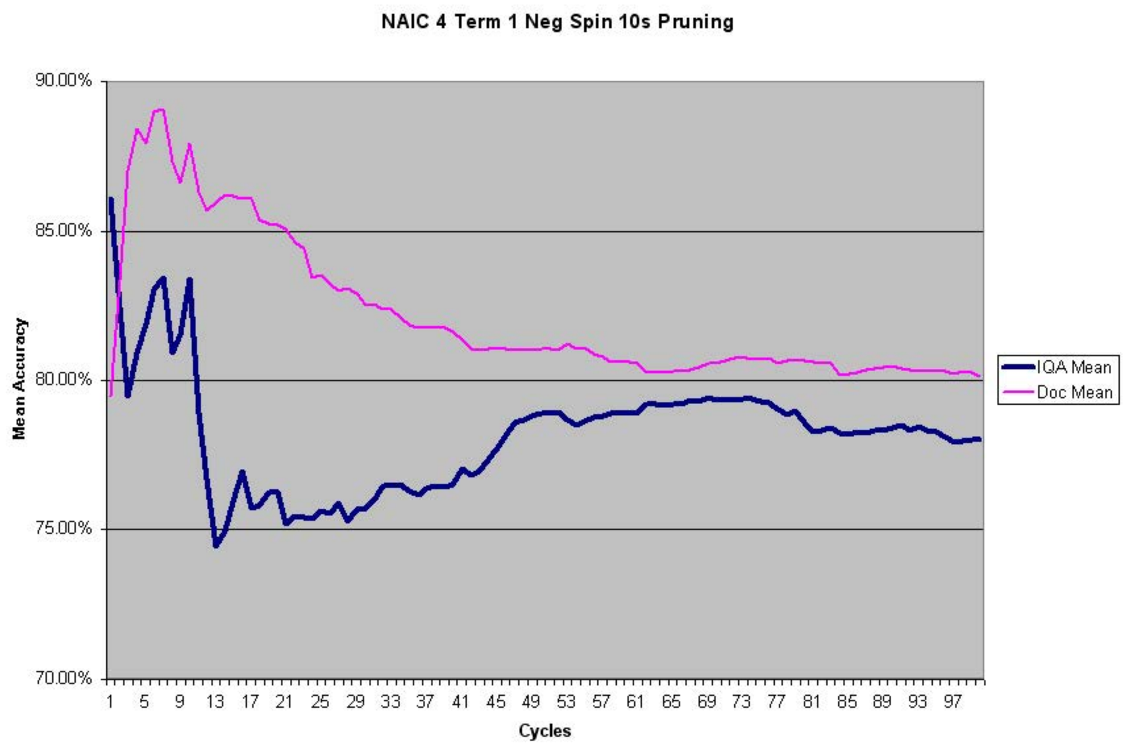

Shape 3 Term 2 Neg Spin 5S Pruning
80.00%
75.00%
70.00%
65.00%
60.00%
Mean Accuracy
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31 33 35 37 39 41 43 45 47 49
Cycles
IQA Mean
Doc Mean

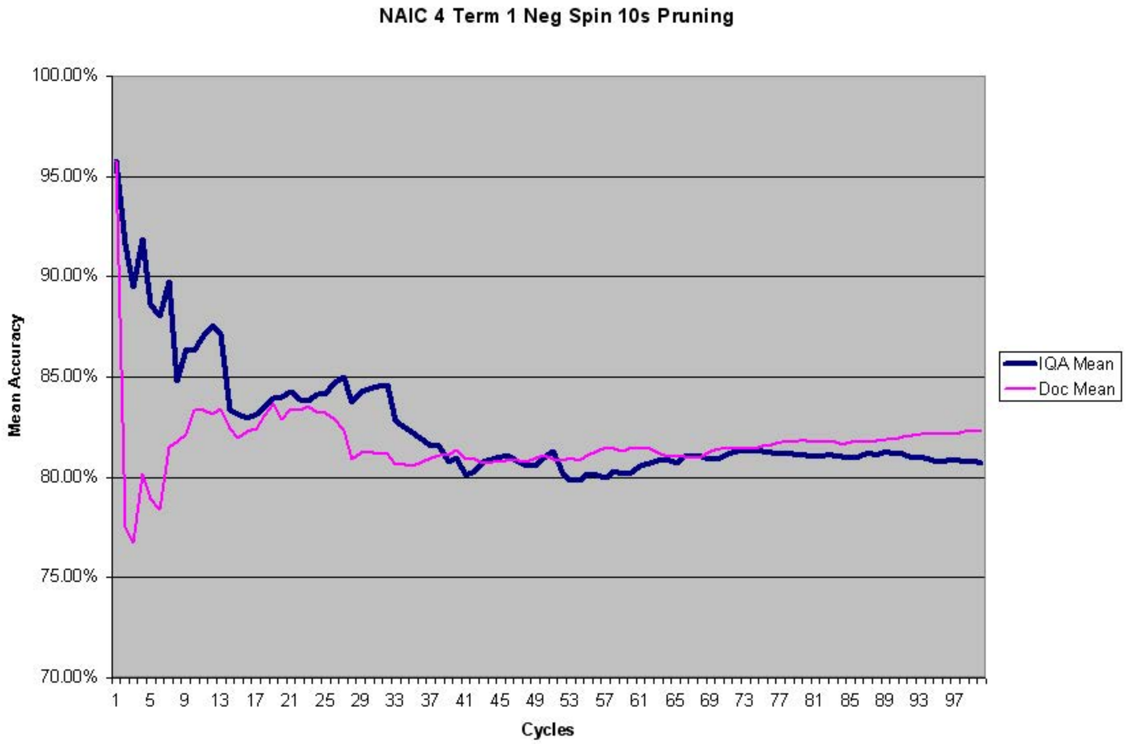
4. 'NAIC Dataset' 4-Term-1



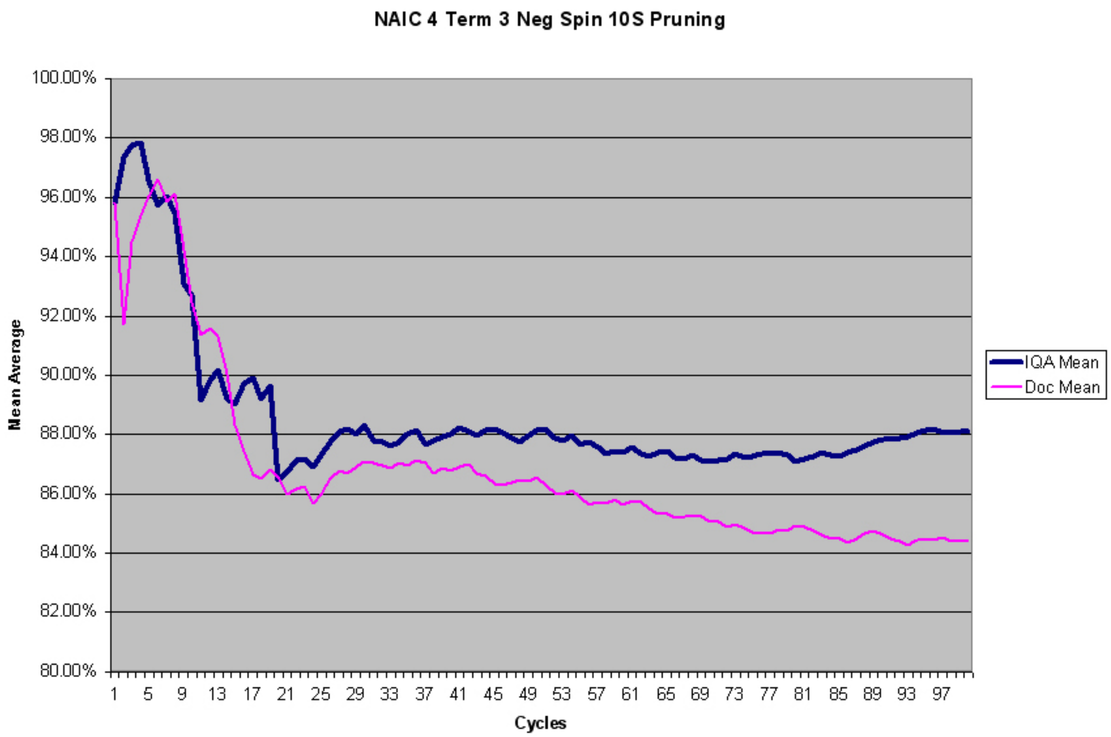


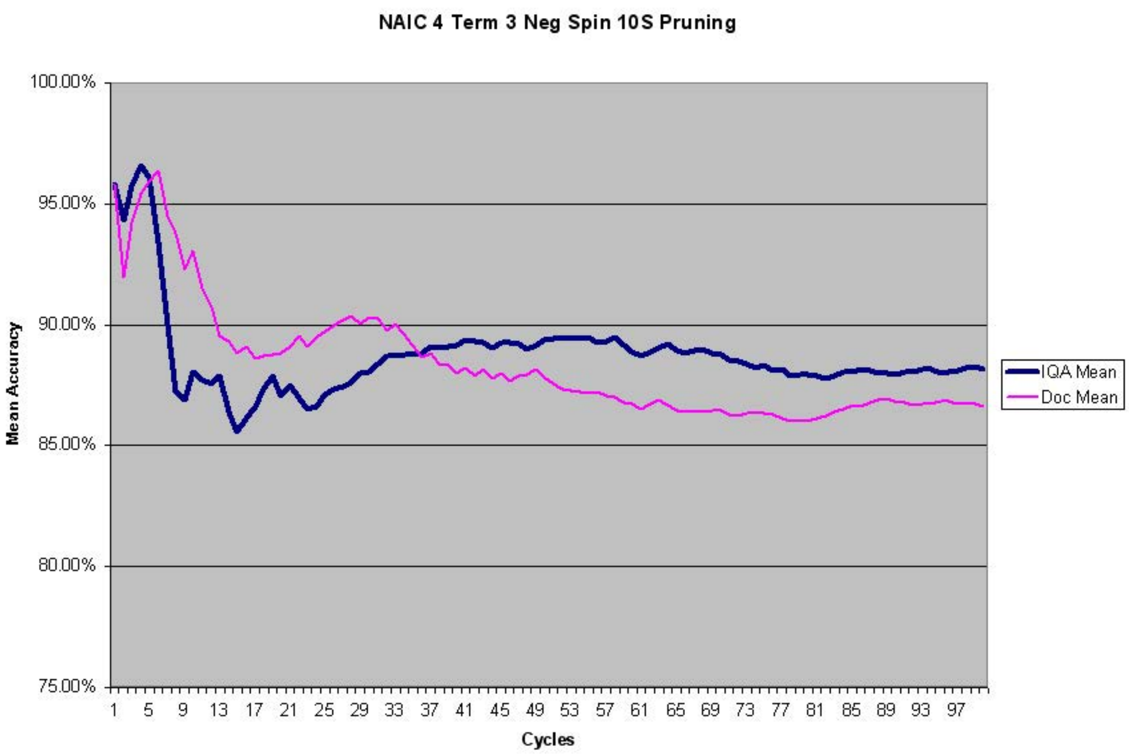

NAIC 4 Term 1 Neg Spin 10s Pruning
100.00%
95.00%
90.00%
85.00%
80.00%
75.00%
70.00%
Mean Accuracy
1 5 9 13 17 21 25 29 33 37 41 45 49 53 57 61 65 69 73 77 81 85 89 93 97
Cycles
IQA Mean
Doc Mean


NAIC 4 Term 1 Neg Spin 10s Pruning
90.00%
85.00%
80.00%
75.00%
70.00%
Mean Accuracy
1 5 9 13 17 21 25 29 33 37 41 45 49 53 57 61 65 69 73 77 81 85 89 93 97
Cycles
IQA Mean
Doc Mean

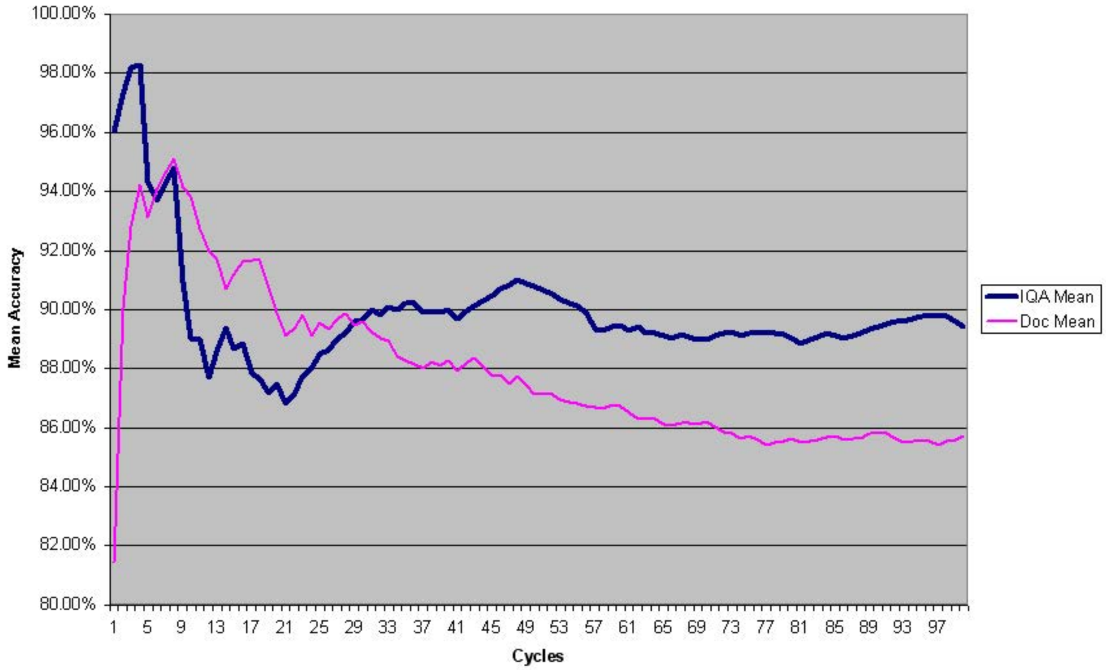

NAIC 4 Term 1 Neg Spin 10s Pruning
100.00%
95.00%
90.00%
85.00%
80.00%
75.00%
70.00%
Mean Accuracy
1 5 9 13 17 21 25 29 33 37 41 45 49 53 57 61 65 69 73 77 81 85 89 93 97
Cycles
IQA Mean
Doc Mean

5. 'NAIC Dataset' 4-Term-3

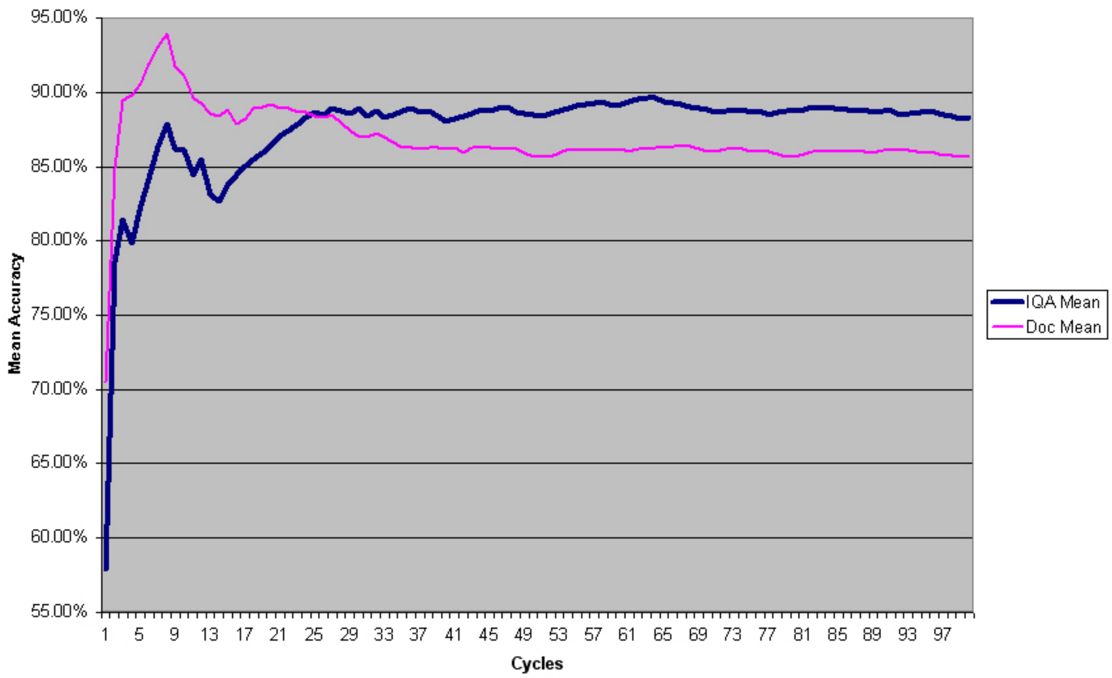


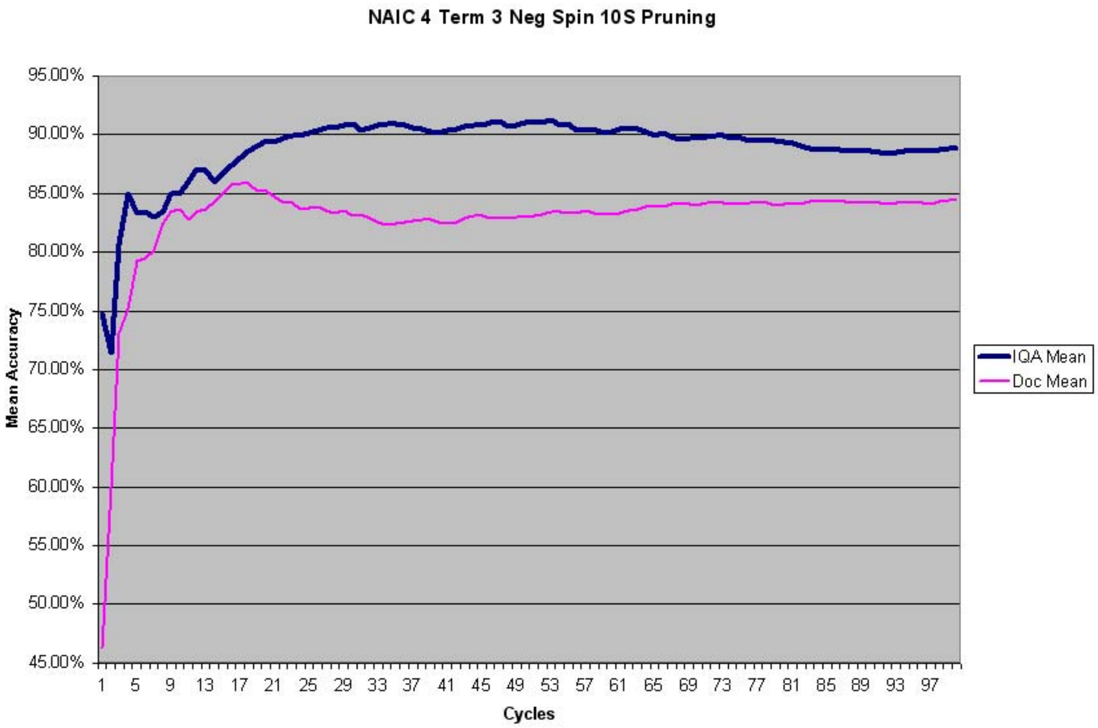


NAIC 4 Term 3 Neg Spin 10S Pruning



NAIC 4 Term 3 Neg Spin 10S Pruning





NAIC 4 Term 3 Neg Spin 10S Pruning
95.00%
90.00%
85.00%
80.00%
75.00%
70.00%
65.00%
60.00%
55.00%
50.00%
45.00%
Mean Accuracy
1 5 9 13 17 21 25 29 33 37 41 45 49 53 57 61 65 69 73 77 81 85 89 93 97
Cycles
IQA Mean
Doc Mean

BIBLIOGRAPHY

[ALL95] Allan, J., “Relevance feedback with too much data,” in *Proc. of the18th annual international ACM-SIGIR conference on research and development in information retrieval*, Seattle, WA, July 1995, pp. 337-343.

[ART99] Arteimi, M, “Rule Refinement In Inductive Knowledge-Based Systems,” Department of Computer Science, Witwatersrand University, Johannesburg, SA,1999.

[BAC03] Backshas, B.K., “Image Retrieval Using A Graphical Human Computer Interface With Feedback-Driven Querying,” Air Force Institute of Technology, 2003.

[BAZ99] Baeza-Yates, R. and Ribeiro-Neto B. *Modern Information Retrieval*. New York: ACM Press and Addison-Wesley Longman Ltd., 1999.

[BLU97] Blum, A., “Empirical support for Winnow and Weighted-Majority algorithms: Results on a calendar scheduling domain,” *Machine Learning*, vol. 26, 1997, pp. 5–23.

[BRE99] Breazeale, D., “The Organization of Internet Web Pages Using WordNet and Self-Organizing Maps,” University of Arlinton, 1999, pp. 9-14.

[BRI98] Brin, S. and Page L., “The anatomy of a large-scale hyper-textual web search engine,” in *Proc. of 7th WWW Conference*, 1998.

[COH92] Cohen, W., Greiner R., and Shuurmans D., “Probabilistic Hill–Climbing,” *Proceedings of Computational Learning Theory and ’Natural’ Learning Systems*, 1992.

[COH95] Cohen, W., “Fast effective rule induction,” in *Proc. of Twelfth International Conference on Machine Learning (ICML-95)*, Morgan Kaufman, 1995, pp. 115–123.

[CRE98] Crestani, F., Lalmas M., Rijsbergen, C.J.V., and Campbell, I., “Is this document relevant?...Probably: A survey of probabilistic models in information retrieval,” *ACM Computing Surveys*, vol. 30, no. 4, Dec. 1998, pp. 528–552.

[DIA03] Diaspie, D. Personal interview at NAIC, 31 July 2003.

[ESP96] Esposito, F., Laterza A., Malerba, D., and Semeraro, G., “Refinement of Datalog Programs,” In *Proceedings of the MLnet Familiarization Workshop on Data Mining with Inductive Logic Programming,”* 73-94, 1996.

[FEL93] Fellbaum, C., Gross, D., and Miller, K. J., (1993). ``Adjectives in WordNet," [Online]. Available: http://www.cogsci.princeton.edu/~wn/index.shtml

[HER93] Hearst, M., and Plaunt, C., "Subtopic Structuring For Full-Length Document Access," in *Proc. of the 16th Annual Intl. ACM SIGIR Conference on Research and Development in Information Retrieval*, Pittsburgh, PA, 1993, pp 59-63.

[JOA97] Joachims, T., "A Probabilistic Analysis of the Rocchio Algorithm with TFIDF for Text Categorization," in *Proc. ICML-97,* 143-146, 1997.

[JON98] Jones, K.S., Walker, S., and Robertson, S.E., "A Probabilistic Model of Information Retrieval: Development and Status." Technical Report TR-446, Cambridge University Computer Laboratory, 1998.

[MAR60] Maron, M.E. and Kuhms, J.L., "On relevance, probabilistic indexing and information retrieval," in *Journal of the ACM,* 7, 1960, pp. 216-244.

[MIL03] Miller, G. A., (1990). "Nouns in WordNet: A Lexical Inheritance System," [Online]. Available: http://www.cogsci.princeton.edu/~wn/index.shtml

[MIL90] Miller, G.A., Beckwith, R., Fellbaum, C., Gross, D., and Miller, K. J.. ``Introduction to WordNet: an on-line lexical database," in *International Journal of Lexicography* 3 (4), 1990, pp. 235-244.

[MIT97] Mitchell, T.M., *Machine Learning.* Boston, MA: McGraw-Hill, 1997.

[MOO95] Mooney. R.J. and Califf, M.E., "Induction of first-order decision lists: Results on learning the past tense of English verbs," in *Proceedings of the 5th International Workshop on Inductive Logic Programming*, Department of Computer Science, Katholieke Universiteit Leuven, 1995 pp. 145–146.

[PAZ91] Pazzani, M, and Brunk, C., "Detecting and Correcting Errors in Rule-Based Expert Systems: An Integration of Empirical and Explanation-Based Learning," *Knowledge Acquisition* 3:2, June 1991, pp. 157-173.

[PAZ92] Pazzani M. and Kibler D., "The Utility of Knowledge in Inductive Learning," *Machine Learning*, 9, 1992, pp.57–94.

[QUI86] Quinlan, J. R., "Induction of decision trees," *Machine Learning*, 1, 1986, pp. 81-106.

[QUI90] Quinlan, J.R., "Learning Logical Definitions From Relations," *Machine Learning*, 5, 1990, pp. 239-266.

[QUI93] Quinlan R.,*C4.5, Programs for machine learning*. Morgan Kaufman, 1993.

[SLA91] Slayton, G., “Developments in Automatic Text Retrieval,” in *Science* (253), 1991, pp. 974-979.

[SPL04] Spliethof, B., “Enhancing Image Retrieval Using Forward Feedback, Term Co-occurrence and Natural Segmentation on a Semi-static Image Collection,” Air Force Institute of Technology, 2003.

[VER00] Versteegen, L., (2000). "The Simple Bayesian Classifier as a Classification Algorithm," [Online]. Available: http://www.cs.kun.nl/nsccs/artikelen/leonv.ps.Z

[WIL03] Wilson, K., “Application of Multi-Modal Feedback Models in Imagery Collections,” Air Force Institute of Technology, 2003.

[WIL94] Wilkinson, R., “Effective retrieval of structured documents,” in *Proc. of the Seventeenth Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval*, Dublin, 1994, pp. 311–317.

[WOR03] Wordnet (2003). [Online]. Available:http://www.cogsci.princeton.edu/cgi-bin/webwn, 2003.

[XIN03] Xin, Q, Miller, E., Long, D., Brandt, S., Schwarz, T., and Litwin, W., “Reliability mechanisms for very large storage systems,” presented at the 20th IEEE/11th NASA Goddard Conference on Mass Storage Systems and Technologies, San Diego, CA, 2003.

VITA

Captain James M. Carsten enlisted in the U.S. Air Force in 1983 and spent 16 years in a variety of enlisted specialties. Those specialties included linguist, dental and program management duties. In 1998, he earned a Bachelor of Science from the University Maryland University College. Capt Carsten received his commission as a Second Lieutenant in May of 1999 after completion of Air Force Officer Training School.

Capt Carsten attended Basic Communication Officer Training en route to Hill Air Force Base. At Hill, he served in many capacities including Officer-In-Charge, Electronics Branch, 75th Communications Squadron; 75th Support Group Executive Officer; and Deputy Chief, Network Management. He deployed in support of Operation Enduring Freedom.

Captain Carsten entered the Air Force Institute of Technology (AFIT) as a graduate student in the Computer Science and Engineering Department. He graduated from AFIT in March of 2004 with a Master of Science degree in Computer Systems, with focuses in Software Engineering and Artificial Intelligence.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 074-0188

1. REPORT DATE *(DD-MM-YYYY)*
23-03-2004

2. REPORT TYPE
Master's Thesis

3. DATES COVERED *(From – To)*
September 2003-March 2004

4. TITLE AND SUBTITLE
INTELLIGENT QUERY ANSWERING THROUGH RULE LEARNING AND GENERALIZATION

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Carsten, James M., Captain, USAF

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(S)
Air Force Institute of Technology
Graduate School of Engineering and Management (AFIT/EN)
2950 Hobson Way, Building 640
WPAFB OH 45433-8865

8. PERFORMING ORGANIZATION REPORT NUMBER
AFIT/GCS/ENG/04-05

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
National Air Intelligence Center
Jeff Null, Chief, Imagery Systems Branch
IMINT Exploitation Division
4180 Watson Way, WPAFB, OH 45433-5648

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

13. SUPPLEMENTARY NOTES

14. ABSTRACT: The Department of Defense (DoD) relies heavily on information systems to complete a myriad of tasks, from day-to-day personnel actions to mission critical imagery retrieval, intelligence analysis, and mission planning. The astronomical growth in size and performance of data storage systems leads to problems in processing the amount of data returned on any given query. Typical relational database systems return a set of unordered records. This approach is acceptable in small information systems, but in large systems, such as military image retrieval systems with more than 1 million records, it requires considerable time (often hours to days) to sort through thousands of records and select the relevant for analysis.

This research introduces Intelligent Query Answering (IQA) as a novel approach to information retrieval. IQA implements the FOIL algorithm to learn rules based upon user feedback [QUI90]. The Winnow algorithm adjusts rule weights based on user classification, for improved document orderings [BLU97]. A semantic tree specific to the domain allows rule generalization across the domain.

Testing shows a document sort accuracy rate of 63-93% against a controlled test dataset and 78-89% accuracy rate on a subset of declassified National Air Intelligence Center imagery metadata. These results demonstrate that this research provides groundwork for future efforts in rule learning and rule generalization in the information retrieval field.

15. SUBJECT TERMS:
FOIL, Rule Learning, Rule Generalization, Winnow, Semantic Tree

16. SECURITY CLASSIFICATION OF: a. REPORT	b. ABSTRACT	c. THIS PAGE	17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON / 19b. TELEPHONE NUMBER *(Include area code)*
U	U	U	UU	141	Gilbert L. Peterson, Ph.D. (937) 255-6565, ext 4281 (gilbert.peterson@afit.edu)