

1-1-2007

## Document type classification from document images

Jason Montgomery Vergara  
*University of Nevada, Las Vegas*

Follow this and additional works at: <https://digitalscholarship.unlv.edu/rtds>

---

### Repository Citation

Vergara, Jason Montgomery, "Document type classification from document images" (2007). *UNLV Retrospective Theses & Dissertations*. 2268.  
<http://dx.doi.org/10.25669/7fm1-s9xr>

This Thesis is protected by copyright and/or related rights. It has been brought to you by Digital Scholarship@UNLV with permission from the rights-holder(s). You are free to use this Thesis in any way that is permitted by the copyright and related rights legislation that applies to your use. For other uses you need to obtain permission from the rights-holder(s) directly, unless additional rights are indicated by a Creative Commons license in the record and/or on the work itself.

This Thesis has been accepted for inclusion in UNLV Retrospective Theses & Dissertations by an authorized administrator of Digital Scholarship@UNLV. For more information, please contact [digitalscholarship@unlv.edu](mailto:digitalscholarship@unlv.edu).

DOCUMENT TYPE CLASSIFICATION FROM DOCUMENT IMAGES

by

Jason Montgomery Vergara

Bachelor of Science  
Whitworth College  
1996

Master of Business Administration  
University of Phoenix  
2004

A thesis submitted in partial fulfillment  
of the requirements for the

**Master of Science Degree in Computer Science  
School of Computer Science  
Howard R. Hughes College of Engineering**

**Graduate College  
University of Nevada, Las Vegas  
December 2007**

UMI Number: 1452280

### INFORMATION TO USERS

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleed-through, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

**UMI**<sup>®</sup>

---

UMI Microform 1452280

Copyright 2008 by ProQuest LLC.

All rights reserved. This microform edition is protected against unauthorized copying under Title 17, United States Code.

ProQuest LLC  
789 E. Eisenhower Parkway  
PO Box 1346  
Ann Arbor, MI 48106-1346

Copyright by Jason Montgomery Vergara 2008  
All rights reserved



**Thesis Approval**  
The Graduate College  
University of Nevada, Las Vegas

OCTOBER 8TH, 2007

The Thesis prepared by

JASON MONTGOMERY VERGARA


Entitled

DOCUMENT TYPE CLASSIFICATION FROM DOCUMENT IMAGES

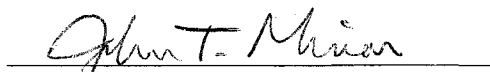
is approved in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

  
Examination Committee Chair

  
Dean of the Graduate College

  
Examination Committee Member

  
Examination Committee Member

  
Graduate College Faculty Representative

## ABSTRACT

### **Document Type Classification from Document Images**

by

Jason Montgomery Vergara

Dr. Kazem Taghva, Examination Committee Chair  
Professor of Computer Science  
University of Nevada, Las Vegas

The most common features that classification systems use is simply to consider all words as features and determine the probability of the document's category based on these words. When given document images, sophisticated optical character recognizers can be used to provide more than the simple text that traditional classification systems use. This metadata and extracting additional features from the document text can improve classification of document images.

We have found a greater than 1% increase in recall when looking at font size metadata and extracting other features such as words used in uppercased lines. Since our dataset can have multi-page documents taking only words on the first page increased recall at least 15%. Approximately 2% of recall was increased by ensuring that 100 words of every document was used; this can be explained by some documents having useless header pages that have very little features.

## TABLE OF CONTENTS

ABSTRACT .....	iii
LIST OF TABLES .....	vi
LIST OF FIGURES .....	vii
ACKNOWLEDGEMENTS .....	viii
CHAPTER 1 INTRODUCTION .....	1
Enterprise Content Management Systems .....	2
Document Management Systems .....	3
This Study .....	4
CHAPTER 2 BACKGROUND .....	6
Single-Label versus Multi-Label Classification .....	7
Binary versus Graded Classification .....	7
Feature Extraction .....	8
CHAPTER 3 METHOD .....	9
Ecdysis .....	10
k-Dependence Algorithm .....	10
Document Dataset .....	12
Evaluation Method .....	12
CHAPTER 4 DOCUMENT TYPE CLASSIFICATION .....	14
Initial Investigations (First Pass) .....	14
Confirmation Investigation (Second Pass) .....	16
Number of Pages and Words .....	17
Ecdysis: Feature Set Limitations .....	18
CHAPTER 5 CONCLUSION AND FUTURE WORK .....	20
Future Work .....	22
APPENDIX A DESCRIPTION OF CLASSIFICATION RUNS .....	24
APPENDIX B RECALL AND PRECISION RESULTS .....	27
APPENDIX C SAMPLE K-DEPENDENCE CALCULATION .....	29
REFERENCES .....	34





## LIST OF TABLES

Table 1	Comparison to the First Document Management System .....	4
Table 2	Document Dataset.....	12
Table 3	Outcomes of Classification .....	13
Table 4	Recall and Precision Varying Number of Pages .....	17
Table 5	Recall and Precision Varying Number of Pages (CR9xxx) .....	18
Table 6	Recall Results Varying FSIZE.....	19
Table 7	Description of Features Sets Investigated and Reported.....	24
Table 8	Recall and Precision Results .....	27

## LIST OF FIGURES

Figure 1	Enterprise Content Management Systems .....	3
Figure 2	Artificial Intelligence.....	6
Figure 3	Preprocessing.....	9
Figure 4	Ecdysis Processing .....	10

## ACKNOWLEDGEMENTS

I would like to thank my colleagues at the Information Science Research Institute (ISRI) for their help, support, and guidance through this thesis and many other projects that we worked on at ISRI. Through this help and support, this thesis would have never come to fruition.

A special thanks to Dr. Kazem Taghva and Jeffrey Coombs who have helped tremendously, especially in covering difficult topics like the k-Dependence Algorithm and Ecdysis. Additional thanks to Steven Lumos, Allen Condit, and Jeffrey Coombs for the development, support, and maintenance of Ecdysis. Thanks to Steven Lumos and Jeffrey Coombs for pointing me in the right direction in learning the ruby programming language. I am grateful to Allen Condit and Jay Nietling for maintaining and supporting the network, workstation, and server resources needed to store and process documents for Ecdysis. Thank you to Julie Borsack for her introduction to ISRI's document type project and Whitney LePore for her quick responses to direct questions I had about document types and specific documents in question.

Last, but not least, a special thanks to my thesis committee, Dr. Ajoy Datta, Ph. D., Dr. Shahram Latifi, Ph. D., Dr. Thomas Nartker, Ph. D., Dr. Kazem Taghva, Ph. D., and Dr. John T. Minor, Ph. D.

## CHAPTER 1

### INTRODUCTION

Periodization divides human history into periods (Webster, 2007); Prehistoric eras such as the Bronze Age and historical periods such as the Renaissance in Europe or the Industrial Revolution in the United States. The last three periods are known as the Information Age, the Knowledge Economy, and the Intangible Economy.

The Information Age lasted approximately twenty years from 1971 through 1991 (Bunch et al., 2004). During this period information technology improved allowing information collections to grow and propagate at higher speeds. Personal computers became more popular in our homes and we have seen electronic communication devices go from 300 baud modems to 10 megabit broadband connections today. This has lead to our society's access to information and the Internet, at home, work, and school.

The Knowledge Economy lasted approximately ten years from 1992 through 2002 (Sipp et al., 2006). In this period businesses become more global, computer networking improves, and 70% of workers are information technology workers; more business transactions are done over computer networks.

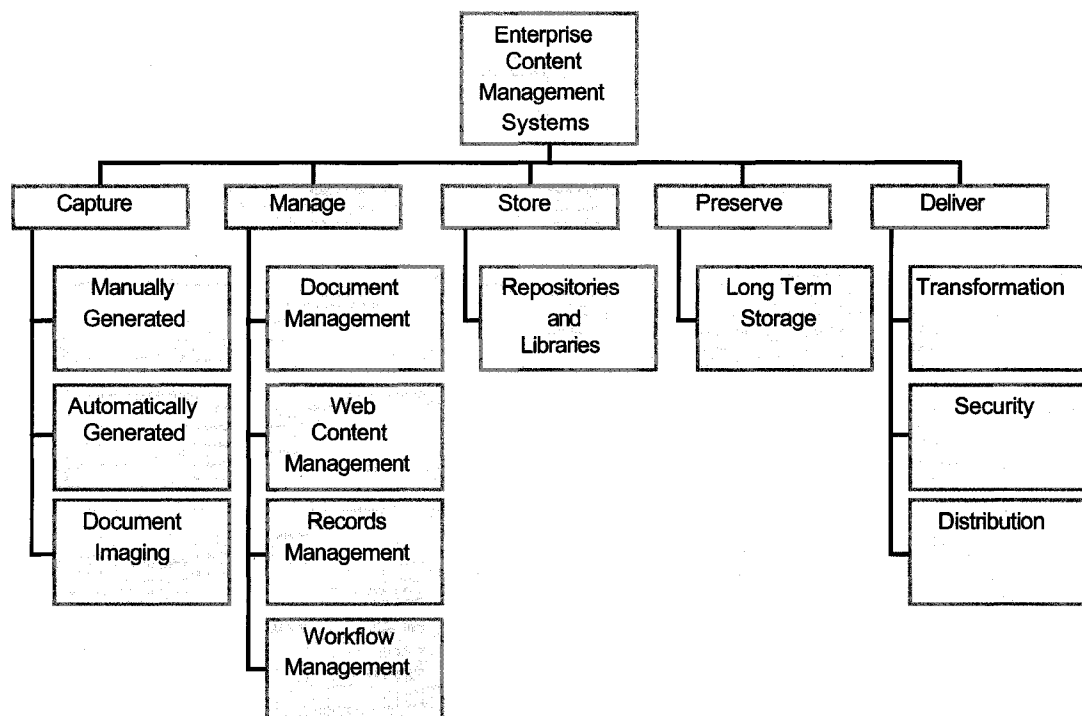
The Intangible Economy started approximately five years ago in 2002 (Andriessen, 2004). Today's economy is not based on physical goods, but virtual non-physical data. In this period business performance is based on intellectual

property and knowledge, and does not depend on your location or the physical resources available to the business (Goldfinger, 2007).

### Enterprise Content Management Systems

Over the last three eras, since the Information Age, information, knowledge, and data collections have grown. Though the Information Age started in 1971 and we have seen technological advances that have contributed to the Information Age at the beginning of this period, information growth and rapid propagation didn't start until the 1980s. For this reason, many companies have developed Enterprise Content Management Systems or simply Content Management Systems. Content Management Systems are used to capture, manage, store, preserve and deliver content (Green, 1993), see Figure 1; often these services also provide revision control, destruction, cataloging/indexing, annotating, and many other important functions needed to manage content. Content is often document images, but can include recorded audio or video, digital photographs, animations, music, web content, and many other forms of digital or digitized content.

Figure 1. Enterprise Content Management Systems (Wikipedia, 2007a)



### Document Management Systems

As mentioned earlier, the Information Age technically started in 1971, but information propagation and growth didn't start until the early 1980s. At this time, companies started developing Document Management Systems to manage paper documents through document imaging. The first Document Management System started off with only a manually indexed storage and retrieval of document images, see Table 1.

Metadata is "data about the data" (Singh, 2005). It describes attributes of the data, in this case a document image. The user determines what Metadata he or she wants or needs to collect about the document image; for example, date/time of storage, text, title, author, date, address, company, number of pages, etc. Metadata can be manually entered by a user or automatically

generated by a computer application. For example, the text of a document can be typed in by a user or automatically entered into the database using an OCR application. From a document image, a user can enter other metadata such as title into the database manually. Once the text is provided for the document, another application can extract the title automatically and enter it into the database.

	First System	Today's Systems
capture	document image, manual metadata	document image, OCR text, automatic metadata extraction, electronic documents (computer files, email, faxes)
manage	index	index, collaboration and workflow tools
distribute	retrieval	retrieval, security, auditing, distribution

### This Study

This paper is about document type classification from document images. The Information Science Research Institute has many projects on metadata extraction from document images. Document images are processed through an OCR application to provide the document text to applications that extract metadata from the text. This project is a study to extract and classify document images to a set of pre-defined document types.

In second section, this study will give a background on classification. The third section will discuss more technical methods on classification for this study

such as Naïve Bayes Classification and k-dependence. The fourth section will present the results of this study and the fifth section will conclude and discuss future work.



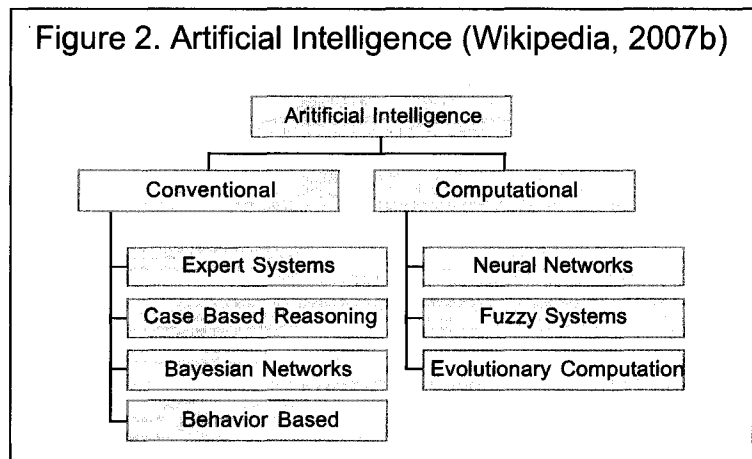
## CHAPTER 2

### BACKGROUND

John McCarthy in 1955 first used the phrase artificial intelligence to mean “the science and engineering of making intelligent machines” (McCarthy, 2004).

There are several areas of Artificial Intelligence, see Figure 2. Conventional artificial intelligence uses formal and statistical methods while computational artificial intelligence uses informal, non-statistical, iterative methods. Machine learning is often associated with conventional artificial intelligence. Each area uses different methods for knowledge acquisition, knowledge storage, and

knowledge retrieval. A complete discussion of the entire artificial intelligence field is beyond the scope of this study.



Conventional artificial intelligence methods are often used in document classification. Expert rule-based systems have rules that look for certain patterns to classify a document into the appropriate category; rules are generally in the form of a condition, for example, “if x and y, then z.” Though the example is simple, an expert system can require many complicated rules. These rule are typically generated by hand. Machine learning methods that use Bayesian and statistical algorithms are also used (Taghva, 2007).

### Single-Label versus Multi-Label Classification

When designing a text classifier, you are given a dataset of documents and given a task to label or classify the documents with a single category or multiple categories. When given  $M$  categories and  $|M| > 1$ , single-label classification requires that the classifier associates the documents in the dataset to exactly one category or label. In effect, the dataset of documents are partitioned and clustered into different, distinct subsets.

Multi-label classification requires that the classifier allow a document to belong to zero, one, all, or some of the  $M$  categories. The result of multi-labeled classification is that each document is associated to a set of categories it belongs to,  $N$ , where  $N \subseteq M$  (McCallum, 1999).

### Binary versus Graded Classification

Classification systems generally are either binary or graded (Tiantian, 2002). Binary classifiers, when given a document, will determine if the document belongs to the category or not. For example, the output of a binary classifier that

determines if a document belongs to the sports category will only be yes, it does or no, it doesn't.

Graded classifiers (Tiantian, 2002), also known as one-of-M classifiers (D'Alessio et al., 2000), when given a document, will determine the rank or degree for the document belonging to the each of the M categories. The category with the larger rank is selected as the category for the document. The output of a graded classifier will often be probabilistic and the highest probable category will be selected.

When classifying documents to more than one category,  $|M| > 1$ , multiple binary classifiers are used to independently determine each of the M categories. This would also allow documents to have zero to M category labels. Graded classifiers on the other hand will be assigned one category depending on which category has the largest rank (D'Alessio et al., 2000). So, binary classifiers seem more useful than graded classifiers in multi-label scenarios. It has been found that graded classifiers are better than binary classifiers when the dataset contains single-labeled documents (D'Alessio et al., 1998).

## Feature Extraction

Text classifiers typically use the document words as features; the document is considered a "bag of words" (Tan, 2000). In terms of word phrases, "bag of words" is a unigram representation of features; Tan's study also looked at two word phrases, bigrams. As we will see later, there are many other features that can be extracted from a document and used for text classification; some examples are symbols, numbers, margin sizes, and so on.

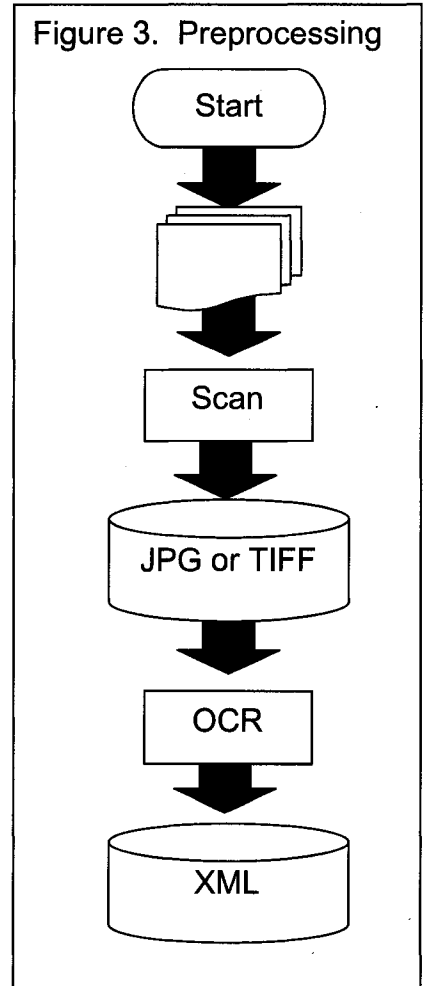
## CHAPTER 3

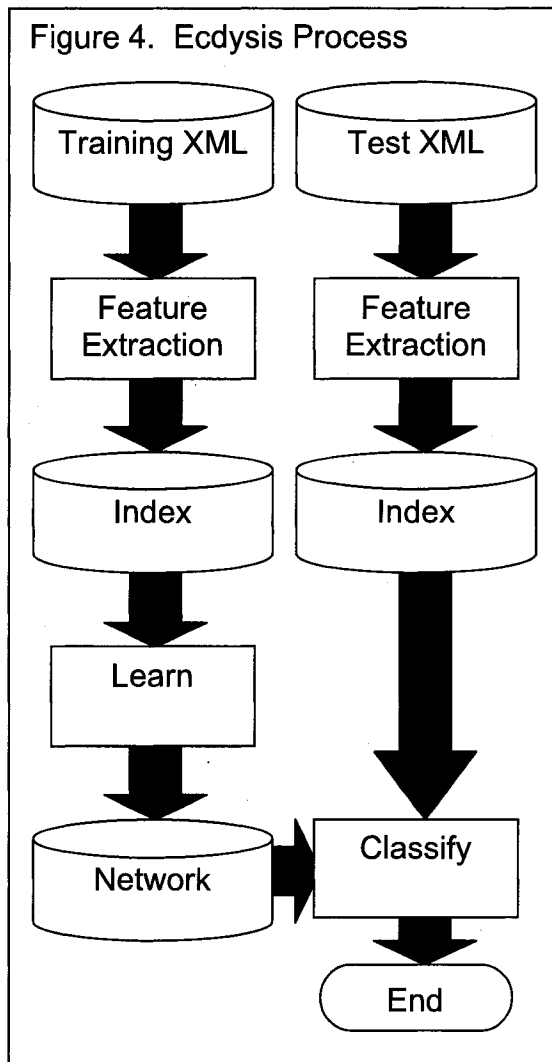
### METHOD

As mentioned earlier, the Information Science Research Institute has many projects on metadata extraction from document images. Document images are processed by an OCR application and from there an hybrid version of an application called Ecdysis extracts features and uses a k-dependence network to classify a document to a particular category.

The pre-processing process starts off with physical documents being scanned in and stored as JPG or TIFF image files. The image files are then processed by the OCR application and stored as an XML file. These XML files contain the words and also other information

about the words and the document itself; for example, word position, word size, word style, document layout, and so on. These XML files represent the documents that are used by Ecdysis.





As mentioned in the previous section XML files are used to represent documents. These documents are manually separated into two groups: training and test. Feature extraction is separately ran on each of the Training XML and Test XML groups to form a separate index for each. From here, the two indexes are treated separately.

First, the training index must be processed by the learning process to produce a network. After the network is produced, the classifier then uses this network to classify the documents

found in the test index. The output of the classifier are the categories for each of the documents that were indexed in the Test XML group.

#### k-Dependence Algorithm

As described in the previous section, Ecdysis produces and uses a network in its learning and classifying processes. A k-dependence algorithm is used to build a Bayesian network. The appendix has a simple example of using the k-dependence algorithm to build the Bayesian network and briefly describes

how it is used in the classification process. In this section, we are going to briefly discuss the algorithm at an abstract level. The algorithm for generating a Bayesian network is (Sahami, 1996):

1.  $\forall X_i$ , compute  $\alpha \leftarrow MI(X_i, C)$
2. Sort and renumber features  $X_1 \dots X_n$  in descending order by  $\alpha$
3.  $\forall i \neq j$ , compute  $\gamma \leftarrow MI(X_i, X_j | C)$
4. for  $i=1..n$  do
  - a.  $r \leftarrow \min(i-1, k)$
  - b.  $parents(X_i) \leftarrow r$  features with largest  $\gamma_{i,j}$  where  $j < i$ .
5. compute the conditional probability tables using the network structure and training set

In the process of using this algorithm, we need to compute Mutual Information (MI) for one and two features,  $MI(X_i, C)$  and  $MI(X_i, X_j | C)$ , respectively. When  $C$  is the set of categories and  $X_i \in \{0, 1\}$  when the feature  $X_i$  is present or not present, one feature mutual information can be computed with the following equation:

$$MI(X_i, C) = - \sum_C P(C) \log_2 P(C) + \sum_{C, X_i} P(C, X_i) \log_2 P(C | X_i)$$

Similarly, two feature mutual information can be computed with the following equation,  $X_j \in \{0, 1\}$ :

$$MI(X_i, X_j | C) = \sum_{X_i, X_j, C} P(X_i, X_j, C) \log_2 \frac{P(X_i, X_j | C)}{P(X_i | C)P(X_j | C)}$$

To avoid introducing zeros into calculations, a special case is introduced by Ecdysis taken from (Kohavi et al., 1997). Where  $n$  is the total number of documents, we replace zero probabilities with:

$$\frac{1/n}{n + 2/n}$$

### Document Dataset

The Information Science Research Institute has a database of labeled documents. The original database consists of a multi-labeled dataset. To simplify the study, we narrowed the dataset to only include single-labeled documents, and from that only took ten categories, see Table 2.

Category	Documents
Calibration	22652
Change	6967
Data	51877
Design	9130
Email	1229
Notebook	2875
Plan	5221
Procurement	2353
Report	32683
Requirement	2595
Total	137582

### Evaluation Method

The classification algorithm will have an output of what category it has computed that the document belongs to. There are four outcomes of the classification when comparing the output to ground truth (the actual category as determined by an expert), see Table 3.

- True Positive (TP): The output **correctly** labeled this document as being in this category.
- False Positive (FP): The output **incorrectly** labeled the document as being in this category.

- False Negative (FN): The output **incorrectly** labeled the document as not being in this category.
- True Negative (TN): The output **correctly** labeled the document as not being in this category.

Output	Ground Truth	¬Ground Truth	Total
Category	True Positive	False Positive	Category
¬Category	False Negative	True Negative	¬Category
Total	Ground Truth	¬Ground Truth	N

The correct outcomes are true positive and true negative – where the output has agreed with the ground truth; these are the numbers we want to maximize. The incorrect outcomes are false positive and false negative; these are the numbers we want to minimize.

Evaluation can be done through recall and precision for each classifier. Recall is the number of documents the classifier has correctly identified as being in that category out of the number of documents the ground truth says is in that category; “out of how many documents in this category did the classifier find.” Precision is the number of documents the classifier has correctly identified in that category out of the number of documents it labeled as being in the category; “out of all the documents the classifier labeled in this category did the classifier label correctly.”

$$recall = \frac{TP}{TP + FN} \quad precision = \frac{TP}{TP + FP}$$



## CHAPTER 4

### DOCUMENT TYPE CLASSIFICATION

The Ecdysis Process outlined in the previous chapter has a long execution time that is primarily because of the indexing phase; during the indexing phase, XML documents are opened and features are extracted. Execution time depends heavily on the number of documents being processed and the size of the documents themselves.

To realize the execution time costs, classification runs CR1, CR13, CR25, and CR37 (described below) took approximately 54, 18, 21, and 33 hours to execute 8% of the entire set of documents. If these runs were on the entire 137,582 documents, an estimated approximation of the execution time would total 57 days. Over fifty different feature set investigations were performed, so running the Ecdysis Process on all documents for each classification run would be prohibitive and impossible.

#### Initial Investigations (First Pass)

To reduce feature investigation and execution time, a smaller, random sample of documents from each category are selected for the feature investigation process. In the first investigation, 1000 documents were selected from each category; a 1-to-1 training-ratio was used where 50% of the selected documents were used for training and 50% were selected for testing. During this

first run, only recall of all categories were used to determine which features remained in the next index. Appendix A, Table 7 has a description of the classification runs, CRs; the first run only included two-digit CRs.

The results of this first run will not be quantitatively reported here. The CRs in bold in Appendix A, Table 7 are classification runs that did not have a decrease in recall; there was an improvement or no change in recall. Here are a few feature sets that have had little or no improvement on recall:

- Floating point and integer feature sets (CR9, CR14, CR15, CR16, CR17)
- Looking at lines that are uppercased or capitalized (CR28)

Feature set classification runs that improved recall:

- Taking at least one page **and** at least 100 (CR20). There are many documents that only have simple pages at the beginning of the document; a page with a “received stamp” or a header page with a few words.
- Many requirement documents contained phrases like “requirement document” or “maintenance requirements” (CR32)
- Adding individual word counts as individual features (CR39)
- Font size matters: Emphasizing above average and large words (CR44)
- Using a traditional stop list (CR45)

There are also some interesting observations that can be made about classification runs that have decreased recall:

- Words containing non-word, decimal, or underscore characters (CR5)
- Using the entire document not only increases execution time and drive space to store the index, it also decreased recall (CR6, CR14)

- Using  $n*n$  not only increases execution time and drive space to store the index, it also severely decreased recall (CR31)

### Confirmation Investigation (Second Pass)

The email category is the smallest category with 1229 documents. For the confirmation investigation, 1229 documents are selected from each category to keep an equal number of documents from each category; a 3-to-1 training-testing ratio is used, where 75% of the selected documents are used for training and 25% are used for testing. The resulting dataset contained 9220 training and 3070 testing documents.

Appendix B, Table 8 contains the precision and recall values for the second pass; again, the second pass only includes two-digit CRs. The confirmation investigation confirms all the decreased recalls observed in the initial investigation. However, the second run only confirms the following recall improvements (improvement must be  $>1.00\%$ ):

- Taking at least one page **and** at least 100 (CR20). There are many documents that only have simple pages at the beginning of the document; a page with a “received stamp” or a header page with a few words.
- Font size matters: Emphasizing above average and large words (CR44)

The second run doesn’t confirm the first run’s improvement of recall of the following CRs:

- Many requirement documents contained phrases like “requirement document” or “maintenance requirements” (CR32)
- Adding individual word counts as individual features (CR39)

- Using a traditional stop list (CR45)

The second pass reveals a new feature set that has improvement (>1.00%):

- Words in uppercased lines (CR34)

### Number of Pages and Words

The same dataset from the confirmation investigation is used here.

CR8xxx and CR9xxx were classification runs to investigate the effect of limiting indexing on the number of pages or words. To isolate pages and words, we first started by limiting pages (CR9xxx) and then limiting by words (CR8xxx). These classification runs are based on CR32; CR32 limits the document indexing to 1 page **and** 100 words.

In CR9xxx runs only the number of pages limit document indexing.

Limiting the document indexing to 2 pages was found to be most optimal for our set of documents (CR9002) when compared to the other page limits. Table 4 shows the recall and precision when varying the number of pages.

Pages	Average		Change	
	Recall	Precision	Recall	Precision
1	77.39%	80.94%	-1.60%	-2.55%
2	77.95%	82.57%	-1.04%	-0.92%
3	75.67%	81.34%	-3.32%	-2.15%
4	74.43%	79.75%	-4.56%	-3.74%
8	70.33%	75.91%	-8.66%	-7.58%
16	65.57%	71.49%	-13.42%	-12.00%
32	64.01%	69.68%	-14.98%	-13.81%

**Note:** The change column is in comparison to CR32

In CR8xxx runs, we keep the 2 page limit and also vary the minimum number of words must be used in document indexing. Limiting the document indexing to 50 words was found to be most optimal for our set of documents (CR8001) when compared to the other word limits. Table 5 shows the recall and precision when varying the number of words.

Words	Average		Change	
	Recall	Precision	Recall	Precision
50	78.14%	82.94%	-0.85%	-0.55%
100	77.20%	81.07%	-1.79%	-2.42%
200	77.75%	82.63%	-1.24%	-0.86%

**Note:** The change column is in comparison to CR32

CR9xxx and CR8xxx conclude that 50 words and 2 pages are most optimal with 78.14% (CR8001). However, CR32 that uses 100 words and 1 page still outperforms CR8001 by 0.85%.

#### Ecydis: Feature Set Limitations

Ecydis has several internal parameters that can be modified to change its behavior. One of these parameters is called FSIZE, feature size. FSIZE limits the number of features that can be used to generate the network used to classify documents. For all classification runs before this, a FSIZE of 512 was used. To reduce investigation time, only four feature sets are used, CR32, CR34, CR39, and CR44. Table 6 shows the results of varying FSIZE.

ID	# of Features	FSIZE				
		512	1024	2048	4096	8192
CR32	55608	78.99%	79.93%	81.53%	82.54%	83.75%
CR34	133243	80.36%	80.42%	81.66%	82.51%	83.88%
CR39	94972	78.79%	<b>76.97%</b>	<b>77.88%</b>	78.89%	80.98%
CR44	106226	80.03%	<b>77.10%</b>	<b>78.40%</b>	<b>79.45%</b>	81.34%

As expected, the results show that recall increases as we increase FSIZE; although there are five runs (in **bold**) that are worse than a smaller FSIZE run using the same feature set. In an ideal world, we could increase FSIZE to match the number of features available. Increasing FSIZE increases execution time; execution time indexing remains the same, but learning and classifying phases increase. Based on CR39, the approximate learning phase takes 1 hour per 1024 features in FSIZE. The approximate classifying phase takes 1 hour per 4096 features in FSIZE.

## CHAPTER 5

### CONCLUSION AND FUTURE WORK

The most common features that classification systems use is simply to consider all words as features and determine the probability of the document's category based on these words. When given document images, sophisticated optical character recognizers can be used to provide more than the simple text that traditional classification systems use. This metadata and extracting additional features from the document text can improve classification of document images.

After two passes looking at different feature sets, we have found a greater than 1% increase in recall when looking at font size metadata and extracting other features such as words used in uppercased lines. Since our dataset can have multi-page documents taking only words on the first page increased recall at least 15%. Approximately 2% of recall was increased by ensuring that 100 words of every document was used; this can be explained by some documents having useless header pages that have very little features.

FSIZE, page limits, and word limits are closely related to the performance of Ecydsis classification. Recall that FSIZE is an internal Ecydsis parameter that limits the number of features that can be used to create the classification network. Ideally, we would want to increase FSIZE to include all possible features. Theoretically, this is also the case for page and word limits. However,

taking all the features from all pages of every document would cause learning and classifying to take extremely long. The performance of Ecydis classification should plateau [way] before all the features from all pages of every document – assuming that there are some less significant features or words.

There are *finite* and *infinite feature sets*. Finite feature sets can only add a finite number of features into the classification system. An example of this would be the email feature used in this study. As long as the indexer detects @ or © after “to:”, “cc:”, or “from:” one feature was added. A count of something can be made finite by setting some kind of limit; an example is CR16 where the number of floating point numbers are separated into a fixed number of groups. Infinite feature sets can add an infinite number of features into a classification system; an example is adding words or large words as features (in reality there is a fixed number of words in any language, but FSIZE is much smaller than the number of words in the English language or in the set of documents).

An argument could be made that the first two classification investigations were unfair comparisons because FSIZE was fixed to 512. CR<sub>x</sub> is the base of CR<sub>y</sub> and CR<sub>y</sub> adds at most z more features. When the features are limited to FSIZE additional features from CR<sub>y</sub> can displace at most z features from CR<sub>x</sub>. The result is an increase or decrease of recall from CR<sub>x</sub> to CR<sub>y</sub>. By running the classification on an FSIZE of 512+z would probably be a much better evaluation of increase or decrease of the new features added to the CR<sub>y</sub> feature set. Another side of this argument is that positive and negative changes in recall or precision don't necessarily say the new features are improvements or not. A 1%



increase is simply better than a 1% decrease; the new features added to the system overpower the features lost by the displacement of features.

### Future Work

There are many investigations that have been started from the main track of this study; all of these investigations were paused to continue other investigations. Here is a brief description of these investigations:

- Portable Stemming and Traditional Stop Lists. Theoretically, this should help improve classification; it has been used in a few other classification systems. Since stemming and stop lists help reduce indexes, more features will be used in the FSIZE limited classification networks.
- Improved Investigations on Finite and Infinite Feature Sets. Currently, we are starting with finite feature sets rather than including an infinite feature set from the beginning (as in this study). After adding an infinite feature set, it would be interesting to look at the effects of increasing FSIZE by the number features added by a finite feature set.
- n-Grams. Other studies have done this for at least 2-grams or bi-grams; rather than looking at only words, two words in sequence are used. There were three investigations started, each with different levels of manual intervention. The first study involved manually opening document images and pulling phrases a human felt were common in that document category. The second study ran tools to look at frequencies of 1- to 5-grams. The third study simply added 1- to 5-grams as features into Ecydsis.

There are several other internal Ecydsis parameters that can be investigated; for example, all classification runs use  $k=2$   $k$ -dependence. There are two other large scale investigations or improvements. The first is to parallelize Ecydsis to make investigations more feasible on multiprocessor servers. All phases, indexing, learning, and classifying can be parallelized. This would help directed feature set investigations with large values of FSIZE.

The second large scale investigation can be to use multiple classification networks, in different typologies. The categories for this dataset are natively multi-labeled. Several studies have claimed that using separate classifiers for each category is the only way to classify multi-labeled documents. For example, if there are  $m$  categories, there would be  $m$  distinct classifiers, one for each category; each classifier will say whether the document is or isn't a member of that category. These separate classifiers form a compound classifier that will output zero to  $m$  categories. There can be many other typologies combining compound classifiers and even using a single-label classifier to ensure there is at least one most probable category.

## APPENDIX A

### DESCRIPTION OF CLASSIFICATION RUNS

Table 7. Description of Features Sets Investigated and Reported		
ID	Base	Description
CR2	N.A.	Entire document. Email-header feature is added if @ or © are found with “to”, “cc”, or “from”. All words are added if it does not contain non-word, decimal, or underscore character.
CR3	N.A.	Entire document. Email features are added if @ or © are found after “to:”, “cc:”, or “from:” on the same line. All words are added if it does not contain a non-word, decimal, or underscore character.
CR4	CR3	Only the first page.
CR5	CR4	All words (even if the word contains a non-word, decimal, or underscore character)
CR6	CR5	Entire Document. Every feature is expanded to include its page. For example, originally “scope” would be a feature. But now, “scope-3” and “scope-10” would be two distinct features for scope appearing on page 3 and 10.
CR7	N.A.	<b>Baseline:</b> Only the first page. “Actual Words” are used; cs.unlv.edu is considered a word in the XML documents. At this point we transition to “cs”, “unlv”, “edu”. There are no e-mail features.
<b>CR8</b>	CR7	Email features are added if @ of © are found after “to:”, “cc:”, or “from:” on the same line.
<b>CR9</b>	CR8	Float count and integer count features are added.
CR14	CR9	All pages.
<b>CR15</b>	CR9	Rather than count, existence is used; 1 if there was a float or integer.
<b>CR16</b>	CR15	Float and integer features are added by power of 2 weight; if there are more than 32 integers, integer-32 is added as a feature, if there are 12 integers, integer-8 is added; similarly with floats

Table 7. Description of Features Sets Investigated and Reported (continued)		
ID	Base	Description
CR17	CR16	If there are more 64 floats, float-2, float-4, ..., float-64 are added; similarly with integers.
CR18	CR17	Not based on the power of 2 weight; if there are 11 floats, float-1, float-2, ..., float-11 are added; similarly with integers.
CR19	CR17	The highest power of 2 weight is 128 for floats and integers.
CR20	CR17	At least one page and at least 100 words
CR21	CR20	Number of lines uppercased, weighted like float and integer
CR22	CR20	Number of lines all_capitalized, weighted like float and integer
CR23	CR20	Number of lines capitalized, weighted like float and integer
CR24	CR23	Number of lines starting with a number, weighted like float and integer
CR25	CR23	Number of words starting with a number, weighted like float and integer
CR26	CR23	Number of lines centered, weighted like float and integer
CR27	CR20	Everything from 21-26
CR28	CR20	Only 21 and 22
CR29	CR20	Punctuation Classes. Number of lines with punctuations “.,:”, class1 (0-2 punctuations), class2 (3-5), class3 (6-10), class4 (>10), number of lines also weighted like float and integer at the end.
CR30	CR28	<b>Useless:</b> added an additional feature if “requirement” was present in the word.
CR31	CR28	<b>Very Costly:</b> $n \times n$ . For all words $i$ and $j$ , where $i \neq j$ , add “ $i-j$ ” as a feature and not “ $j-i$ ”.
CR32	CR28	Add features for “requirements document”, “requirements matrix”, “assurance requirements”, “equipment requirements”, “installation requirements”, “operational requirements”, “maintenance requirements”, “utility requirements”, “system requirements”
CR33	CR28	Not specific phrases. “ <i>anyword</i> requirements” and “requirements <i>anyword</i> ” are added as features
CR34	CR32	Words that are in uppercased lines are added as special features

Table 7. Description of Features Sets Investigated and Reported (continued)		
ID	Base	Description
CR35	CR32	Words that are in all _capitalized lines are added as special features
CR36	CR32	Words that are in capitalized lines are added as special features
CR37	CR35	The special features are also numbered sequentially to give order
CR38	CR32	Using word count rather than existence for the feature's value
<b>CR39</b>	CR32	Using word-count as a feature; if there are 12 "scope" words, "scope-12" is added as a feature
CR40	CR39	If the first page is less than 100 words, 100 words is used. Now we use the entire last page that the 100 <sup>th</sup> word lies on.
CR41	CR39	nwords/4 added as a feature
CR42	CR39	number of words uppercased, weighted like float and integer
CR43	CR39	number of words capitalized, weighted like float and integer
<b>CR44</b>	CR43	Font size. Features are added for large words. Large words are defined as words above the average of the above average words.
<b>CR45</b>	CR44	Using a traditional stoplist of common words.
CR46	CR34	Font size. Features are added for large words. Large words are defined as words above the average of the above average words.
CR8001	CR9002	At least two pages and at least 50 words
CR8002	CR9002	At least two pages and at least 100 words
CR8003	CR9002	At least two pages and at least 200 words
CR9001	CR32	First page only
CR9002	CR32	First two pages
CR9004	CR32	First four pages
CR9008	CR32	First eight pages
CR9016	CR32	First sixteen pages
CR9032	CR32	First thirty-two pages

## APPENDIX B

### RECALL AND PRECISION RESULTS

ID	Base	Average		Change	
		Recall	Precision	Recall	Precision
CR2	N.A.	58.21%	64.47%	N.A.	N.A.
CR3	N.A.	61.89%	66.83%	N.A.	N.A.
CR4	CR3	77.20%	80.95%	15.31%	14.12%
CR5	CR4	75.86%	79.63%	-1.34%	-1.32%
CR6	CR5	64.56%	67.54%	-11.30%	-12.09%
CR7	N.A.	77.10%	80.99%	N.A.	N.A.
<b>CR8</b>	CR7	77.10%	80.96%	0.00%	-0.03%
CR9	CR8	77.20%	81.07%	0.10%	0.11%
CR14	CR9	64.40%	68.27%	-12.80%	-12.80%
<b>CR15</b>	CR9	77.20%	81.07%	0.00%	0.00%
<b>CR16</b>	CR15	77.10%	80.96%	-0.10%	-0.11%
<b>CR17</b>	CR16	77.17%	80.98%	0.07%	0.02%
CR18	CR17	76.38%	79.75%	-0.79%	-1.23%
CR19	CR17	77.17%	80.98%	0.00%	0.00%
<b>CR20</b>	CR17	79.06%	83.54%	1.89%	2.56%
CR22	CR20	78.96%	83.45%	-0.10%	-0.09%
CR23	CR20	79.02%	83.60%	-0.04%	0.06%
CR24	CR20	78.86%	83.31%	-0.20%	-0.23%
CR25	CR20	78.40%	83.12%	-0.66%	-0.42%
CR26	CR20	78.76%	83.30%	-0.30%	-0.24%
CR27	CR20	77.75%	82.32%	-1.31%	-1.22%
<b>CR28</b>	CR20	78.76%	83.40%	-0.30%	-0.14%
CR29	CR20	78.53%	82.91%	-0.53%	-0.63%
CR30	CR28	78.86%	83.41%	0.10%	0.01%
CR31	CR28	52.02%	56.03%	-26.74%	-27.37%
<b>CR32</b>	CR28	78.99%	83.49%	0.23%	0.09%
CR33	CR28	78.76%	83.41%	0.00%	0.01%
CR34	CR32	80.36%	83.69%	1.37%	0.20%
CR35	CR32	78.86%	83.32%	-0.13%	-0.17%
CR36	CR32	79.15%	83.66%	0.16%	0.17%
CR37	CR32	79.32%	83.41%	0.33%	-0.08%

ID	Base	Average		Change	
		Recall	Precision	Recall	Precision
CR38	CR32	78.99%	83.49%	0.00%	0.00%
<b>CR39</b>	CR32	78.79%	83.16%	-0.20%	-0.33%
CR40	CR39	74.92%	80.68%	-3.87%	-2.48%
CR41	CR39	74.92%	80.68%	-3.87%	-2.48%
CR42	CR39	78.86%	83.04%	0.07%	-0.12%
CR43	CR39	79.02%	83.28%	0.23%	0.12%
<b>CR44</b>	CR43	80.03%	83.51%	1.01%	0.23%
<b>CR45</b>	CR44	79.09%	84.29%	-0.94%	0.78%

## APPENDIX C

### SAMPLE K-DEPENDENCE CALCULATION

In these sample calculations, we will set  $k=2$ , use two categories  $C=\{C_1, C_2\}$ , four features  $X=\{X_1, X_2, X_3, X_4\}$ , and five training documents:

Doc	Category	$X_1$	$X_2$	$X_3$	$X_4$
1	$C_1$	1	0	1	1
2	$C_1$	1	0	1	0
3	$C_1$	1	0	0	0
4	$C_2$	1	1	1	1
5	$C_2$	0	1	0	1

#### Mutual Information calculation for one feature:

$$MI(X_i, C) = -\sum_C P(C) \log_2 P(C) + \sum_{C, X_i} P(C, X_i) \log_2 P(C | X_i)$$

First Term of  $MI(X_i, C)$ :

$$\begin{aligned} & -\sum_C P(C) \log_2 P(C) \\ & = -(P(C_1) \log_2 P(C_1) + P(C_2) \log_2 P(C_2)) = -\left(\frac{3}{5} \log_2 \frac{3}{5} + \frac{2}{5} \log_2 \frac{2}{5}\right) \\ & = -(-0.4422 - 0.5288) = 0.9710 \end{aligned}$$

Special Case Calculation (we use this value to avoid zero calculations):

$$\frac{1/n}{n + 2/n} = \frac{1/5}{5 + 2/5} = \frac{1/5}{27/5} = \frac{1}{27}$$

Second Term of  $MI(X_i, C)$ , where  $i=1$ :



$$\begin{aligned}
& \sum_{C, X_1 \in \{0,1\}} P(C, X_1) \log_2 P(C | X_1) \\
&= P(C_1, X_1 = 0) \log_2 P(C_1 | X_1 = 0) + P(C_1, X_1 = 1) \log_2 P(C_1 | X_1 = 1) \\
&\quad + P(C_2, X_1 = 0) \log_2 P(C_2 | X_1 = 0) + P(C_2, X_1 = 1) \log_2 P(C_2 | X_1 = 1) \\
&= \frac{1}{27} \log_2 \frac{1/27}{1/5} + \frac{3}{5} \log_2 \frac{3/5}{4/5} + \frac{1}{5} \log_2 \frac{1/5}{1/5} + \frac{1}{5} \log_2 \frac{1/5}{4/5} \\
&= -0.0901 - 0.2490 + 0.0000 - 0.4000 = -0.7391
\end{aligned}$$

Finally, summing the two intermediate values results in  $MI(X_1, C)$ :

$$MI(X_1, C) = 0.9710 - 0.7391 = 0.2318$$

→ This calculation is performed for all features in  $X$ ,  $X_1$  to  $X_4$ .

**Mutual Information calculation for two features:**

$$MI(X_i, X_j | C) = \sum_{x_i, x_j, C} P(X_i, X_j, C) \log_2 \frac{P(X_i, X_j | C)}{P(X_i | C)P(X_j | C)}$$

We also use  $1/27$  to prevent calculations with zero, here is a sample calculation

for  $i=1$  and  $j=2$ .

$$\begin{aligned}
MI(X_1, X_2 | C) &= \sum_{x_1, x_2, C} P(X_1, X_2, C) \log_2 \frac{P(X_1, X_2 | C)}{P(X_1 | C)P(X_2 | C)} \\
&= P(X_1 = 0, X_2 = 0, C_1) \log_2 \frac{P(X_1 = 0, X_2 = 0 | C_1)}{P(X_1 = 0 | C_1)P(X_2 = 0 | C_1)} \\
&\quad + P(X_1 = 0, X_2 = 1, C_1) \log_2 \frac{P(X_1 = 0, X_2 = 1 | C_1)}{P(X_1 = 0 | C_1)P(X_2 = 1 | C_1)} \\
&\quad + P(X_1 = 1, X_2 = 0, C_1) \log_2 \frac{P(X_1 = 1, X_2 = 0 | C_1)}{P(X_1 = 1 | C_1)P(X_2 = 0 | C_1)} \\
&\quad + P(X_1 = 1, X_2 = 1, C_1) \log_2 \frac{P(X_1 = 1, X_2 = 1 | C_1)}{P(X_1 = 1 | C_1)P(X_2 = 1 | C_1)} \\
&\quad + P(X_1 = 0, X_2 = 0, C_2) \log_2 \frac{P(X_1 = 0, X_2 = 0 | C_2)}{P(X_1 = 0 | C_2)P(X_2 = 0 | C_2)} \\
&\quad + P(X_1 = 0, X_2 = 1, C_2) \log_2 \frac{P(X_1 = 0, X_2 = 1 | C_2)}{P(X_1 = 0 | C_2)P(X_2 = 1 | C_2)} \\
&\quad + P(X_1 = 1, X_2 = 0, C_2) \log_2 \frac{P(X_1 = 1, X_2 = 0 | C_2)}{P(X_1 = 1 | C_2)P(X_2 = 0 | C_2)} \\
&\quad + P(X_1 = 1, X_2 = 1, C_2) \log_2 \frac{P(X_1 = 1, X_2 = 1 | C_2)}{P(X_1 = 1 | C_2)P(X_2 = 1 | C_2)}
\end{aligned}$$

$$\begin{aligned}
&= \frac{1}{27} \log_2 \frac{(1/27)/(3/5)}{((1/27)/(3/5))(3/3)} \\
&+ \frac{1}{27} \log_2 \frac{(1/27)/(3/5)}{((1/27)/(3/5))((1/27)/(3/5))} \\
&+ \frac{3}{5} \log_2 \frac{(3/5)/(3/5)}{(3/3)(3/3)} \\
&+ \frac{1}{27} \log_2 \frac{(1/27)/(3/5)}{(3/3)((1/27)/(3/5))} \\
&+ \frac{1}{27} \log_2 \frac{(1/27)/(2/5)}{(1/2)((1/27)/(2/5))} \\
&+ \frac{1}{5} \log_2 \frac{(1/5)/(2/5)}{(1/2)(2/2)} \\
&+ \frac{1}{27} \log_2 \frac{(1/27)/(2/5)}{(1/2)((1/27)/(2/5))} \\
&+ \frac{1}{5} \log_2 \frac{(1/5)/(2/5)}{(1/2)(2/2)} \\
&= 0 + \frac{1}{27} \log_2 \frac{81}{5} + 0 + 0 + \frac{1}{27} \log_2 2 + 0 + \frac{1}{27} \log_2 2 + 0 \\
&= 0.1488 + 0.0370 + 0.0370 = 0.2229
\end{aligned}$$

→ This calculation is performed for all pairs of features in X.

The following table lists all the mutual information values for one and two features:

		M(X <sub>i</sub> , X <sub>j</sub>   C)			
	M(X <sub>i</sub> , C)	x <sub>1</sub>	x <sub>2</sub>	x <sub>3</sub>	x <sub>4</sub>
x <sub>1</sub>	0.2318	0.4427	0.2229	0.3742	0.1544
x <sub>2</sub>	0.6950	0.2229	0.2760	0.1544	0.2075
x <sub>3</sub>	0.1370	0.3742	0.1544	0.7079	0.1936
x <sub>4</sub>	0.2928	0.1544	0.2075	0.1936	0.5412

The algorithm sorts the features by one feature mutual information,

$M(X_i, C)$ :

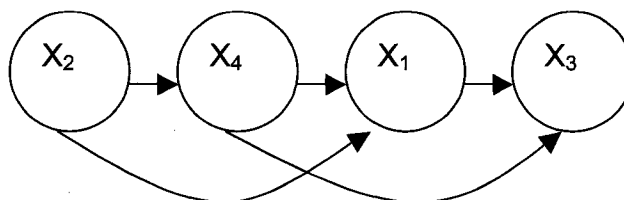
		$M(X_i, X_j   C)$			
	$M(X_i, C)$	$x_2$	$x_4$	$x_1$	$x_3$
$x_2$	0.6950	0.2760	0.2075	0.2229	0.1544
$x_4$	0.2928	0.2075	0.5412	0.1544	0.1936
$x_1$	0.2318	0.2229	0.1544	0.4427	0.3742
$x_3$	0.1370	0.1544	0.1936	0.3742	0.7079

Looking closer at the algorithm, you can see that after sorting based on  $M(X_i, C)$ ,  $r$  parents are selected by the following two criteria:

- (1) the parent must have been already added to the graph, and
- (2) which features the term has the greatest two feature mutual information values

$i$	$r$	term	possible parents	selected parents
1	0	$x_2$	none	none
2	1	$x_4$	$x_2$	$x_2$
3	2	$x_1$	$x_2 x_4$	$x_2 x_4$
4	2	$x_3$	$x_2 x_4 x_1$	$x_1 x_4$

The graph is generated:



Each node in the network will contain a table of probabilities for all the categories with respect to its parent nodes. For example,  $X_3$  could have the following probabilities:

$P(X_3   X_1, X_4)$				
$X_3$	$X_1$	$X_4$	$C_1$	$C_2$
0	0	0	0.0	0.0
0	0	1	1.0	1.0
0	1	0	0.0	0.0
0	1	1	0.0	0.3
1	0	0	0.0	0.0
1	0	1	0.0	0.7
1	1	0	0.0	0.6
1	1	1	0.0	0.0

After the learning process is complete, the network is formed and all the probabilities for each node's table is calculated. A new document is classified using this network. If a new document is to be classified that has the feature vector of (1,0,1,1), the category with the largest probability of the following equation will be selected during classification:

$$\begin{aligned}
 &P(X_1 = 1, X_2 = 0, X_3 = 1, X_4 = 1) \\
 &= P(X_2 = 0) \cdot P(X_4 = 1 | X_2 = 0) \\
 &\cdot P(X_1 = 1 | X_2 = 0, X_4 = 1) \cdot P(X_3 = 1 | X_1 = 1, X_4 = 1)
 \end{aligned}$$

## REFERENCES

- Andriessen, D. Making Sense of Intellectual Capital: Designing a Method for the Valuation of Intangibles. Elsevier Inc.: Burlington, Massachusetts. 2004.
- Bunch, B. and Hellemans, A. The History of Science and Technology: A Browser's Guide to the Great Discoveries Inventions, and People Who Made Them from the Dawn of Time to Today. Houghton Mifflin Company: New York, New York. 2004.
- Carayannis, E. and Sipp, C. e-Development toward the Knowledge Economy: Leveraging Technology, Innovation and Entrepreneurship for 'Smart' Development. Palgrave Macmillan: New York, New York. 2006.
- D'Alessio S., Kershenbaum A., Murray K., Schiaffino R. Category Levels in Hierarchical Text Categorization. Proceedings of the Third Conference on Empirical Methods in Natural Language Processing (EMNLP-3). 1998.
- D'Alessio S., Murray K., Schiaffino R. , Kersehnbaum A. The Effect of Using Hierarchical Classifiers in Text Categorization. New Rochelle, New York: Iona College. 2000.
- Green, W. Introduction to Electronic Document Management Systems. Academic Press Inc.: San Diego, California. 1993.
- Goldfinger, C. What is the New Economy? Retrived from the World Wide Web: <http://www.gefma.com/Intangible.htm>. 2007.

- Kohavi, R., Becker, B., and Sommerfield, D. Improving Simple Bayes.  
Proceedings of ECML-97. 1997.
- McCallum, A. Multi-label Text Classification with a Mixture Model Trained by EM.  
AAAI 99 Workshop on Text Learning. 1999.
- McCarty, J. What is Artificial Intelligence? Retrieved from the World Wide Web:  
<http://www-formal.stanford.edu/jmc/whatisai/whatisai.html>. 2007.
- Miriam-Webster, Incorporated. Periodization. Retrieved from the World Wide  
Web: <http://www.webster.com/dictionary/periodization>. 2007.
- Sahami, M. Learning Limited Dependence Bayesian Classifiers. Second  
International Conference on Knowledge Discovery in Databases. 1996.
- Singh, M. The Practical Handbook of Internet Computing. CRC Press LLC:  
Boca Raton, Florida. 2005.
- Taghva, K., Condit, A., Lumos, S., Borsack, J., and Nartker, T. Title Extraction  
and Generation from OCR'd Documents. 2007.
- Tan, C. Finding and Using High Quality Word-Pairs for Enhanced Text  
Categorization. Santa Barbara, California: University of California, Santa  
Barbara. 2000.
- Tiantian, J. Applying Machine Learning Algorithms to Text Categorization.  
Montreal, Quebec, Canada: McGill University. 2002.
- Wikipedia. Enterprise Content Management. Retrieved from the World Wide  
Web: [http://en.wikipedia.org/wiki/Enterprise\\_content\\_management](http://en.wikipedia.org/wiki/Enterprise_content_management).  
2007a.

Wikipedia. Artificial Intelligence. Retrieved from the World Wide Web:

[http://en.wikipedia.org/wiki/Artificial\\_intelligence](http://en.wikipedia.org/wiki/Artificial_intelligence). 2007b.

## VITA

Graduate College  
University of Nevada, Las Vegas

Jason Montgomery Vergara

Local Address:

1070 Legato Drive  
Las Vegas, Nevada 89123

Home Address:

1070 Legato Drive  
Las Vegas, Nevada 89123

Degrees:

Bachelor of Science, Computer Science, 1996  
Bachelor of Arts, Mathematics, 1996  
Whitworth College

Master of Business Management, Technology Management, 2004  
University of Nevada, Las Vegas

Thesis Title: Document Type Classification from Scanned Images

Thesis Examination Committee:

Chairperson, Dr. Kazem Taghva, Ph. D.  
Committee Member, Dr. Ajoy K. Datta, Ph. D.  
Committee Member, Dr. Thomas Nartker, Ph. D.  
Graduate Faculty Representative, Dr. Shahram Latifi, Ph. D.